Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Ana Liz Souto Oliveira de Araújo

Quantifying Computational Thinking Abilities

Campina Grande – PB
2019

Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Coordenação de Pós-Graduação em Ciência da Computação

Quantifying Computational Thinking Abilities

Ana Liz Souto Oliveira de Araújo

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Educação em Ciência da Computação

Dr. Dalton Dario Serey Guerrero (Orientador)

Dr. Wilkerson de Lucena Andrade (Orientador)

Campina Grande, Paraíba, Brasil

A663q Araújo, Ana Liz Souto Oliveira de.
Quantifying computational thinking abilities / Ana Liz Souto Oliveira de Araújo. – Campina Grande, 2019.
97 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.
"Orientação: Prof. Dr. Dalton Dario Serey Guerrero, Prof. Dr. Wilkerson de Lucena Andrade".
Referências.

1. Ciência da Computação – Estudo e Ensino. 2. Pensamento Computacional. 3. Avaliação de Habilidades Cognitivas. 4. Competição Bebras. I. Guerrero, Dalton Dario Serey. II. Andrade, Wilkerson de Lucena. III. Título.

CDU 004(07)(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

"QUANTIFYING COMPUTATIONAL THINKING ABILITIES"

ANA LIZ SOUTO OLIVEIRA DE ARAÚJO

TESE APROVADA EM 29/08/2019

DALTON DARIO SEREY GUERRERO, Dr., UFCG
Orientador(a)

WILKERSON DE LUCENA ANDRADE, Dr., UFCG
Orientador(a)

LÍVIA MARIA RODRIGUES SAMPAIO CAMPOS , Dra., UFCG
Examinador(a)

JORGE CESAR ABRANTES DE FIGUEIREDO, Dr., UFCG
Examinador(a)

EDUARDO HENRIQUE DA SILVA ARANHA, Dr., UFRN
Examinador(a)

ROBERTO ALMEIDA BITTENCOURT, Dr., UEFS
Examinador(a)

CAMPINA GRANDE - PB

Para as mulheres que me inspiram:
minha filha Giovanna, meu raio de sol;
minha mãe Jailma, meu porto seguro;
minhas avós Raimunda e Liu (in memorian),
marcas de amor e ternura.

Resumo

Pensamento Computacional (PC) consiste em um processo de resolução de problemas explorando habilidades cognitivas do campo da Ciência da Computação. Na última década, diversas abordagens têm sido criadas para disseminar PC, bem como, formas de avaliá-la em estudantes. Assim, vários instrumentos têm sido criados para mensurar habilidades do PC. Na maioria dos casos, esses instrumentos são baseados em programação, como jogos, *surveys* e entrevistas sobre projetos de programação. Na maioria dos casos, esses instrumentos são baseados em programação de computadores, como questionários, pesquisas e entrevistas sobre projetos de programação. Essas abordagens são úteis para medir as habilidades de resolução de problemas dos alunos por meio de atividades de programação. Portanto, esses métodos dependem de atividades de programação para avaliar as habilidades de PC, ou seja, os alunos devem aprender programação de computadores e, em seguida, uma abordagem qualitativa ou quantitativa avalia as habilidades de PC exploradas nas atividades de programação. Em alguns cenários, no entanto, seria essencial ter meios para medir essas habilidades sem exigir conhecimentos prévios de programação dos sujeitos, porque Pensamento Computacional diz mais sobre as habilidades cognitivas usadas para resolver problemas do que sobre habilidades técnicas.

O problema central da área consiste que as estratégias para avaliar as habilidades de pensamento computacional são incipientes. Embora várias abordagens tenham sido propostas para disseminar PC, pouco se sabe sobre como medir as habilidades de maneira confiável. De fato, vários estudos foram publicados recentemente, mas poucos abordam evidências empíricas guiadas por uma metodologia robusta e ferramentas estatísticas. Nosso principal objetivo é investigar estratégias e instrumentos para quantificar as habilidades de PC de maneira confiável, sem práticas de programação obrigatórias. Abordamos esse objetivo considerando as habilidades da PC como variáveis latentes (construtos) exploradas pelos itens, ou seja, perguntas, nos instrumentos.

Durante esta pesquisa, propusemos um modelo teórico de PC baseado em estudos empíricos que podem ser usados para avaliar as habilidades sem práticas de programação obrigatórias. Avaliamos itens projetados para explorar as habilidades de PC usando uma abordagem psicométrica. Essa estratégia nos fornece uma melhor compreensão do que devemos observar para avaliar a PC, bem como uma perspectiva adequada para medir as habilidades da PC. Finalmente, reunimos um conjunto de lições aprendidas sobre as características dos itens, a fim de auxiliar pesquisas futuras para medir as habilidades de PC de

maneira confiável. Em resumo, observamos que é possível quantificar as habilidades da PC usando uma metodologia e procedimentos estatísticos adequados.

Abstract

Computational Thinking (CT) focuses on the problem-solving process using cognitive abilities in the Computer Science field. In the last decade, several approaches have been created to disseminate CT as well as to assess the development of CT skills in students. Therefore, various instruments have been used to evaluate CT abilities. In most cases, these instruments are based on computer programming, such as quizzes, surveys, and interviews about programming projects. These approaches are useful for measuring students' problem-solving skills by means of programming activities. So, these methods depend on programming activities to assess CT abilities, i.e. the students must learn computer programming, and then, a qualitative or quantitative approach assesses the CT abilities exploited in their programming activities. In some scenarios, however, it would be essential to have means to measure such skills without requiring previous programming knowledge from the subjects, because CT is more about the cognitive abilities used to solve problems than about technical skills.

The problem is that the strategies for evaluating computational thinking skills are incipient. Although several approaches have been proposed to disseminate CT, little is known about how to measure CT skill reliably. There has been little quantitative analysis of quantifying CT skills. In fact, various studies have been published recently, but few tackle empirical evidence guided by a robust methodology and statistical tools. Our main objective is to investigate strategies and instruments in order to quantify CT skills reliably without mandatory programming practices. We approach this objective considering the CT skills as latent variables (constructs) explored through the items, i.e. questions, in the instruments.

During this research, we have proposed a theoretical CT model based on empirical studies which can be used for assessing CT skills without mandatory programming practices. We have evaluated items designed to explore CT skills using a psychometric approach. This strategy provides us a better understanding of what we should observe to assess CT as well as a proper perspective for measuring CT abilities. Finally, we have gathered a set of lessons learned on providing the items' characteristics in order to assist future research in design items to measure CT skills reliably. In summary, we have observed that it is possible to quantify CT abilities using a suitable methodology and statistical procedures.

Agradecimentos

Momento de agradecer as pessoas que fizeram parte neste percurso. Meus orientadores, Dalton e Wilkerson, por acreditarem no meu trabalho como pesquisadora e apoiarem minhas decisões. A Monilly, pelo feliz encontro e coorientação nesta pesquisa interdisciplinar. Minha admiração aos profissionais dedicados que vocês são e minha gratidão pelo que aprendi com vocês. Agradeço a banca avaliadora, professores Lívia, Jorge, Roberto e Eduardo, pela disponibilidade de contribuir com minha pesquisa.

Rodrigo, meu marido, por estar ao meu lado desde o início da minha jornada profissional. A minha filha, Giovanna, que me deu força e coragem para terminar esta etapa e começar uma nova versão (melhor) de mim.

Minha mãe Jailma e meu pai Moacir, por terem plantado, desde cedo, a semente que estudar é o melhor caminho para crescer. Por me incentivarem a cada dia ir mais longe. Ao meu irmão Ramon, por ser sempre presente para me escutar e aconselhar. Ainda por nos presentear com Miguel, meu sobrinho amado, seu sorriso foi luz para alegrar meus dias de estudo.

Apesar do caminho solitário que é o percurso do doutorado, fui agraciada com amizades preciosas. Mirna, foi muita sorte minha te encontrar e podermos partilhar as incertezas e sucessos de trilhar uma pesquisa de doutorado (com muito bom humor). Eliane, obrigada pela acolhida no laboratório e por todo apoio. Isabelle, sua parceria e amizade foram fundamentais no decorrer das pesquisas. Jucelio, obrigada por cada troca de conhecimento e referências. Lilian, obrigada pela acolhida no SPLab, estando sempre disponível para ajudar no que fosse preciso.

A professora Valentina Dagiene, da Universidade de Vilnus, por ter colaborado com a pesquisa relacionada ao Bebras.

Contents

List of Symbols

AIC - *Akaike's Information Criterion*

CFA - *Confirmatory Factor Analysis*

CFI - *Comparative Fit Index*

CS - *Computer Science*

CT - *Computational Thinking*

CTT - *Classical Test Theory*

BIC - *Bayesian information criterion*

EFA - *Exploratory Factor Analysis*

FA - *Factor Analysis*

ICC -*Item Curve Characteristic*

IRT - *Item Response Theory*

KMO - *Kaiser-Meyer-Olkin test*

MSA - *Measure of Sampling Adequacy*

PCA - *Principal Component Analysis*

RMSEA - *Root Mean Square Error of Approximation*

SRMR - *Standardized Root Mean Square Residual*

TLI - *Tucker-Lewis Index*

UFCG - *Universidade Federal de Campina Grande*

UFPB - *Universidade Federal da Paraíba*

UK - *The United Kingdom*

1PL - *One-Parameter Logistic Model*

2PL - *Two-Parameter Logistic Model*

3PL - *Three-Parameter Logistic Model*

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, Computational Thinking is introduced as the main research topic. A brief contextualization on measuring Computational Thinking abilities is presented in Section 1.1. The research problem is discussed in Section 1.2. The objectives and the research questions are presented in Section 1.3. The contribution of this doctoral research is declared in Section 1.4. Finally, an overview of the organization of the thesis is presented in Section 1.5.

1.1 Contextualization

Computational Thinking (CT) has gained increasing attention over recent years in Computer Science Education. Computational thinking is a way for humans to solve problems [87]. This statement emphasizes that CT encompasses all mental tools that reflect and support the breadth of the Computer Science field. Thus, CT is more about the cognitive abilities used to settle problems within Computer Science than about technical skills.

The concept of CT is in construction involving several and different skills. Bar and Stephenson have listed the core CT concepts and capabilities for K-12, which comprises data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization and simulation [8]. Grover and Pea have also considered CT abilities including abstraction and pattern generalizations; systematic processing of information, symbol systems, and representations; algorithmic notions of flow of control; structured problem decomposition; iterative, recursive, and parallel thinking; conditional logic; efficiency and performance constraints; debugging and systematic error detection [40]. Because of the broad set of abilities associated with CT, defining the core of CT skills depends on each study.

In our research, CT is a strategy to solve problems exploring skills in Computer Science [87; 88; 19]. We claim that CT skills are more about cognitive abilities than technical skills [78]. Initially, we have adopted the CT skills proposed by Selby in her Ph.D. thesis [83]. Selby has proposed a CT taxonomy which consists of abstraction, algorithmic thinking, decomposition, evaluation, and generalization (including pattern recognition). At the beginning of this study, we have chosen this approach because it expresses the main general CT skills, regardless of the practice approach. However, during this research, we could visualize the CT skills in a different perspective, as we will discuss in the following chapters.

Moving on to a practice perspective, in the last decade, several approaches have been created to disseminate CT, mainly involving programming activities. These approaches have involved teaching programming within diverse languages, such as Scratch [1], App Inventor [2], Alice [3], and Python, as well as programming games using Kodu [4], Scalable Game Design [52].

Another initiative to disseminate CT is Bebras Challenge [5] . Bebras is an international effort that builds on the idea that teaching computer programming directly is not mandatory to stimulate CT. Bebras is a global challenge whose goal is to promote Computer Science and CT among primary and secondary school students, and also for the general public [23]. The contest is a test composed of multiple choice questions and/or short answer as well as interactive tasks related to problem-solving that do not depend on programming.

Although several approaches have been proposed to disseminate CT, little is known about how to measure CT skills reliably [55; 48]. This occurs due to the challenging of assessing a construct or latent trait [76]. In a psychometric perspective, a construct is a cognitive variable that cannot be measurable by direct observation, like time or temperature [47]. It is needed a suitable instrument capable of assessing a particular construct. Then, the results of the instrument should be analyzed by rigorous methodology and statistical resources [7].

[1] Scratch. Available on: https://scratch.mit.edu/

[2] App Inventor. Available on: http://appinventor.mit.edu/explore/

[3] Alice. Available on: https://www.alice.org/

[4] Kodu. Available on: https://www.kodugamelab.com/

[5] Bebras Challenge. Available on: http://www.bebras.org/

The *skill* and *ability* words are used often as a synonym in CT contexts. In fact, when we have searched on the dictionary for the meaning of these words, they seem to be equivalent. The Longman dictionary [6] says that *skill* is "an ability to do something well, especially because you have learned and practised it". Actually, based on the same dictionary, *ability* is "the state of being able to do something", or "someone's level of skill at doing something". Therefore, we have adopted *skill* and *ability* as a synonym in this study.

1.2 Problem Statement

Numerous studies in the last decade shed light on propagating CT practices [55; 48; 51; 50]; nonetheless, there continues to be a gap in the understanding and assessing CT skills. The main problem is that there has been little quantitative analysis of quantifying CT abilities. In fact, various studies have been published recently, but little tackle empirical evidence guided by a robust methodology and statistical tools [48]. As a consequence, the studies may not be replicated, and the allegation may not be supported scientifically. It may seem that the CT research field did not make inroads in producing accurate scientific studies with quality and reproducibility [24; 62].

Beyond the general problem, we face other limitations within the existing proposals. At first glance, computer programming seems to be a perfect approach to understand and measure CT [24], but it has several limitations. Firstly, in most cases, the students need to learn computer programming, and then write codes in order to solve the problems. The drawback of this approach is the CT abilities are assessed based on the programming skills of the subjects. It is necessary to consider that the students are using a new way to express the solution different from their natural language [84; 37; 28; 39; 29]. More than that, we can bias the construct which we want to measure by teaching what we desire to observe. Secondly, the assessment itself often consists in verifying whether the code produced by the students contains certain structural elements from a predefined checklist [80]. Third, measuring CT abilities by code can mix cognitive skills with technical skills. In other words, the struggle of learning a new programming language can influence their ability to express the solution, and consequentially, the students' CT abilities [34].

[6] Longman Dictionary of Contemporary English. Available on: https://www.ldoceonline.com/dictionary

Using tests with coding problems to assess CT abilities is not a better option. Tests are instruments designed to measure the respondents' knowledge. However, when the test is about programming, we come back to the same point above, i.e., the student must learn computer programming in order to measure CT abilities. Besides, the test depends on a programming language, so it can be only used by who teaches the same language [93]. Otherwise, the test's result may be influenced by the methodology of the course, duration time, and previous programming experiences by students [90; 17]. While these approaches are useful for measuring students' problem-solving skills, they must require the subjects to know computer programming.

Using tests without coding problems to assess CT abilities is an incipient way to Computer Science Educational field. There is no validated and widely accepted approach to guide the creation of tests for evaluating CT abilities. Most of the existing tests have been built in an ad-hoc way [27]. They have been developed and used for a particular study without prior planning about the validity of the test. In most of the cases, the tests are not reused, which means that each study creates another new test. However, there are few reliable instruments, but they are designed for specific public and language, such as Spanish students between 12 and 14 years [76] and undergraduate students in Turkey [53]. Because of that, it is challenging to draw a dissimilarity among the effect of different approaches to promote CT or quantify CT abilities themselves.

All these drawback considered, if we have continued to believe that programming is the only choice to understand and measure CT, we will never understand the real cognitive abilities involved in CT. On the other hand, if we have produced instruments to measure CT without rigorous methodology, we not only may produce biased results but also we may not know what it is measured in fact [78]. Therefore, quantifying CT abilities themselves remain unresolved in Computer Science Education as well as how to design trustful instruments.

1.3 Objective and Research Questions

Our main objective is to investigate strategies and instruments in order to quantify CT skills reliably without mandatory programming practices. We approach this objective considering the CT skills as latent variables (constructs) explored through the items, i.e. the questions in the instruments. Considering that items are the central unit of every instrument, we focus on investigating items instead of the entire instrument.

Our proposal is to present a theoretical CT model based on empirical studies which can be used for assessing CT skills. We intend to examine items which explore CT skills using a psychometric approach to gather information which abilities (and how) these items have explored CT abilities.

In this doctoral study, the main research question (RQ) is: *Which strategies and instruments can we use in order to quantify CT skills reliably without mandatory programming practices?* In order to further understand whether it is possible to quantify CT abilities reliably and how, this research focuses on the following specific questions within the broader scope of assessing CT skills:

1. Which skills can we observe in CT?

   We settle this RQ using Factor Analysis. Factor analysis (FA) is a multivariate statistical technique which is suitable for examining the structure and interrelations of variables. Factors represent constructs, i.e., the underlying dimensions of a set of observed variables. In our study, the variables are the items (questions) in the instrument, and the factors have been explored as CT skills.

2. How have items been designed to evaluate CT skills?

   We address this RQ using Item Response Theory. Item Response Theory (IRT) is a mathematical model which is based on individuals' performances on a test designed to measure particular abilities. We used IRT for examining the difficulty level of the items and the discrimination, i.e., how well an item can differentiate students with higher and lower levels of knowledge in CT.

3. What are the practices we can use to produce items to assess CT?

   We address this RQ gathering a set of lessons learned during the conducted research. A critical perspective of the empirical results enables us to propose a new CT model suitable to measure CT skill without programming practices.

1.4 Contributions

The main contribution of this thesis is a new theoretical CT model based on empirical studies. This model can be used for assessing CT skills without mandatory programming practices. We have evaluated strategies and instruments to intend to quantify CT skills as a latent trait. The adopted methodology and the experimental observation assist us in understanding the nature of the cognitive abilities involved in solving problems in the context of CT and how to assess them. Excluding programming practices from our study provides us a view of the high level of competencies in CT. This approach is possible considering the adopted methodology, which helps us rethink what is explored in CT. In summary, the contributions we have addressed in this thesis are:

1. A theoretical CT model that encompasses skills and competencies expressing different granularity.

2. An analysis of how to identify constructs in CT instruments.

3. An evaluation of items designed to explore CT skills. This strategy provides us a better understanding of what we should observe to assess CT as well as a proper perspective for measuring CT abilities.

4. A set of lessons learned on providing items' features intended to assist future research in design reliable items to measure CT skills.

1.5 Thesis' Outline

The outline of this thesis is as follows.

Chapter 1 have described contextualization of CT briefly and highlighted the problem concerning the assessed methods. We have defined our objective and research questions. Finally, we have highlighted our main contributions.

Chapter 2 presents the background contents in order to understand what Computational Thinking consists of and which theories and statistical procedures support our thesis.

Chapter 3 proposes the theoretical model of CT and specifies the adopted strategies of quantifying CT abilities. It explains the limitation of the other theoretical models and how we have organized the CT skill and competencies. It sums up the roadmap of this research and presents the methodology which we have followed to evaluate our claims.

Chapter 4 describes how we investigate CT skill. It details the results of the investigation on how many CT abilities we can observe in the instruments.

Chapter 5 shows how we evaluate CT items. It summarizes the results of the items' characteristics which have influenced in assess CT skills.

Chapter 6 reports a discussion about the findings and understanding of quantifying CT skills. We discusses some threats to validity our study.

Chapter 7 contrasts our research with related works.

Finally, **Chapter 8** summarizes what was done in our doctoral research. In addition, it addresses future works and open research questions which might be carried on in order to extend the assessment of CT skills.

Chapter 2

Background

This chapter presents the background of the thesis. Computational Thinking is discussed in Section 2.1, more precisely, the definitions, the involved abilities and the approaches that promote CT. In addition, we also explain the importance of Bebras Challenge as an instrument to disseminate CT in Section 2.2. We also detail how Bebras community works to maintain the international contest and to increase the research about CT.

To further clarify methodological concepts, the second part of the Chapter explains three theories adopted to guide our analysis. Factor Analysis, Network Analysis, and Item Response Theory assist us in understanding the CT skills and sustaining our claims. The theories are explained in Sections 2.3, Sections 2.4 and 2.5, respectively.

2.1 Computational Thinking

Computational Thinking (CT) was originated by Seymour Papert in the context of his constructionist learning ideas [65] [66]. In 2006, Jannette Wing, in her seminal paper, reintroduces CT as a “fundamental, not rote skills” for problem-solving "builds on the power and limits of computing processes" [87](page 1). Both Wing and Papert emphasize reasoning, but there is a difference between these two initial definitions. The Wing’s definition emphasized problem-solving, and Papert’s definition is more focused on ideas and analysis for stimulating students to think reflectively [60]. Since then, CT has been explored in the field of Computer Science Education research.

The definition of CT is under construction. According to Hu, CT represents a cognitive process; hence, it should be seen as a hybrid paradigm that accommodates different thinking models such as logical, algorithmic, analytical, mathematical, engineering, and creative thinking [44]. Wing says that "Computational Thinking is the thought processes involved in formulating problems and their solutions are represented in a way that is effectively carried out by an information-processing agent" (page 1) [87]. Korkmaz *et al.* emphasize that CT is the ability to explore the knowledge, skills, and attitudes using basic concepts of Computer Science under life's problems [53]. Therefore, it seems that different definitions of CT are acceptable. However, the broad conception does not help in practice. Because of that, it is necessary to specify which skills are involved.

Several research have focused on defining the core set of skills which characterize CT. Wing claims that abstraction is the essence of CT because working with multiple layers of abstraction helps people to understand the problem and solution [88]. Bar and Stephenson have listed the core CT concepts and capabilities for K-12 which comprise data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization and simulation [8]. Grover and Pea have also considered CT abilities for K-12 widely. They included abstractions and pattern generalizations; systematic processing of information, symbol systems and representations; algorithmic notions of flow of control; structured problem decomposition; iterative, recursive, and parallel thinking; conditional logic; efficiency and performance constraints; debugging and systematic error detection [40]. These several abilities demonstrate that there are not well defined or explicit boundaries of CT.

The International Society for Technology in Education [1] (ISTE) and Computer Science Teachers Association [2] (CSTA) have also developed an operational definition for K-12 education. For them, CT is a problem-solving process within the following characteristics [49] (page 1):

- "Formulating problems in a way that enables us to use a computer and other tools to help solve them;
- Logically organizing and analyzing data;

[1] ISTE. Available on: https://www.iste.org
[2] CSTA. Available on: https://www.csteachers.org/

- Representing data through abstractions such as models and simulations;
- Automating solutions through algorithmic thinking;
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;
- Generalizing and transferring this problem solving process to a wide variety of problems".

They also endorsed the CT operational definition with a set of attitudes that includes [49] (page 1):

> *(i)* "Confidence in dealing with complexity; *(ii)* Persistence in working with difficult problems; *(iii)* Tolerance for ambiguity; *(iv)* The ability to deal with open ended problems; and *(v)* The ability to communicate and work with others to achieve a common goal or solution".

Various proposals to promote CT have been created for different audiences. Programming using visual language is the most common approach to stimulate K-12 students for many reasons [24]. For instance, the unnecessary syntax is reduced (e.g., the use of the semicolon and curly brackets), the commands are closer to the natural language spoken, and the students need only to drag and drop the command blocks are some reasons why programming with blocks is popular in CT [59; 76]. Programming activities with robotic is also a pedagogical approach to stimulate CT [54; 63]. Unplugged activities have been used to introduce Computer Science content and CT [75]. On the other hand, some initiatives have also been created for training no-computer science teachers in stimulating CT in K-12. Google for education[3] has created the Exploring Computational Thinking (ECT). ECT is a site which presents a collection of lesson plans, videos, and other resources on CT for teachers. The aim of ECT is to provide a better understanding of CT for educators and administrators, and to support those who want to integrate CT into their classroom content, teaching practice, and learning.

[3] Google for education. Available on: https://edu.google.com/resources/programs/exploring-computational-thinkin

In this study, we adopted the CT definition and abilities proposed by Selby in her Ph.D. thesis [83]. CT is a problem-solving approach which is supported by a set of cognitive abilities, such as abstraction, algorithmic thinking, decomposition, evaluation, and generalization (including pattern recognition). Although Selby's work was related to programming practices, her conceptual model of skill is suitable for no-programming approaches, like our research, because it describes the high level of cognitive abilities. In the following, those abilities are presented in more details.

Abstraction is related to a way of paying attention to the essential information to solve a problem. During the resolution, we ignore what it is not useful [18]. A map for tourists is an example of abstraction. It represents the main tourist attractions and ignores what is not interested in visitors. The map highlights the localization of parks, museums, churches, gardens, and at the same time ignore other buildings that can not be interesting for tourists. In addition, abstraction is also about choosing a representation of a situation [22] to reduce the complexity of a problem [18]. Because of this benefits, abstraction is one of the most important [88] and assessed CT abilities [24].

Algorithmic thinking can be understood as developing sequences and rules intend to reach the final goal [18]. When we are exploring the ability to think like an algorithmic, understanding, creating, and executing are the expected cognitive abilities. [22]. For instance, learning the steps for doing multiplication or division in mathematics is an example of a basic algorithm that all students should learn at school. This example reinforces the idea that computer scientists do not only use algorithms but also apply it in day-life [81]. Algorithmic thinking is also related to abstraction regarding a way to represent one solution. Wing argued that "an algorithm is an abstraction of a step-by-step procedure for taking input and producing some desired output" [88] (p.3718). Namely, algorithmic thinking is an essential and one of the most assessed CT ability, as abstraction [24].

Decomposition can be understood as the ability to break problems down into small and manageable parts to solve them [18]. It is also useful in complex tasks to make them feasible to solve [81]. For instance, organizing a party can be decomposed into distinct activities, like reserving a location, inviting guests, ordering foods, and buying beverages. These activities can also be broken down into smaller activities and be integrated later. Therefore, a complex task can be better managed if we used decomposition.

Evaluation can be seen as achieving the best solution to a problem taking into account the resources. This process of evaluating may include a set of questions, as suggested by Csizmadia *et. al.* [18] page 7: "Are the properties of the solution correct? Are they fast enough? Do they use resources economically? Are they easy for people to use? Do they promote an appropriate experience?". Thus, those questions assist us in finding a solution which suits our purpose [22].

Generalization is the ability to recognize patterns on how to solve problems and apply the previous solution to similar problems [81]. Particularity, generalization involves identifying similarities and connections among activities. For example, if we have to organize a wedding, we can start from a list of activities created when we organized a party because we recognize the similarities between these two events. Generalization also includes induction, e.g., using one general solution to accomplish the activity [22].

2.2 Bebras Challenge

Bebras is an international effort that builds on the idea that teaching computer programming is not mandatory to stimulate CT. Bebras is a worldwide challenge whose goal is to promote Computer Science and CT among primary and secondary school students, and also for the general public [23]. Bebras has been arranged annually since 2004. In 2018, more than more than 2.7 million students from 54 countries[4] participated in the contest.

The central part of the Bebras challenge is a contest runs annually one or two times for six different age groups. The name and grades of the group can vary among countries, but in most of the cases are Pre-Primary (grades 1-2), Little Beavers (grades 3- 4), Benjamin (grades 5-6), Cadet (grades 7-8), Junior (grades 9-10), Senior (grades 11-12). Frequently, the contest is performed in schools using computers but also can be played with paper and pencil. The participants are supervised by teachers who may integrate the competition into their teaching activities[23]. Pupils must solve 15 to 18 tasks within 40 to 55 minutes.

Another part of the Bebras Challenge is the organizing of the contest. Once a year, researchers from all the countries meet in a workshop to design and improve the **tasks** (name of the questions in Bebras Challenge). The tasks are created to be short, answerable in a few

[4] https://www.bebras.org/?q=statistics

minutes through a digital interface, and requiring deep-thinking skills in the informatics field [21]. The tasks are either multiple-choice (four-choice questions with one correct answer) or interactive (using drag-and-drop techniques, assembling constructions, picking items, writing, etc.) [20]. The tasks are created in English. Each national organizer needs to translate the tasks from English into the national language spoken in his country.

Although the test can be answered without prior knowledge about computer programming and computer science in general, all questions are indirectly related to fundamental computational concepts. The categories of contents have changed for Bebras tasks over the years. Nowadays, there are two-dimensional categorization [22]. The first is the categories of Informatics Concepts (knowledge level) which include five topics namely *(i)* Algorithms and programming, including logical reasoning; *(ii)* Data, data structures and representations (includes graphs, automaton, data mining); *(iii)* Computer processes and hardware (includes anything to do with how the computer works – scheduling, parallel processing); *(iv)* Communications and networking (includes cryptography, cloud computing); *(v)* Interaction (Human-Computer Interaction, HCI), systems and society (all other topics). The second is the categories of Computational Thinking (skills level) which include five abilities such as *(i)* Abstraction, *(ii)* Algorithmic thinking, *(iii)* Decomposition, *(iv)* Evaluation, and *(v)* Generalization. This categorization system incorporates both computational thinking skills and Computer Science concepts in the classification of tasks.

Another category of tasks is the difficulty level. The tasks are organized into three categories which present increasing difficulties. Furthermore, each task of a category receives different points. Tasks of the "A" category earn +6 points for a correct answer and 0 points for an incorrect answer; tasks of the "B" category receive +9 points for a correct answer and -2 points for a wrong answer; at last, tasks of the "C" category receive +12 points for a correct answer and -4 points for an incorrect response (see Table 2.1). The stakeholders make this classification by their intuition [85; 86]. Each age group has tasks into those three categories of difficulties.

To illustrate, Figure 2.1 is a example of Bebras' tasks named "Space Maze". This task was classified as an easy question ("A" category) by the UK organizers [43]. They also associated algorithmic thinking as a mandatory skill to solve this task.

Table 2.1: Scores of Bebras Challenge

Difficulty	Easy	Medium	Hard
Correct answer	6	9	12
No answer	0	0	0
Incorrect answer	0	-2	-4

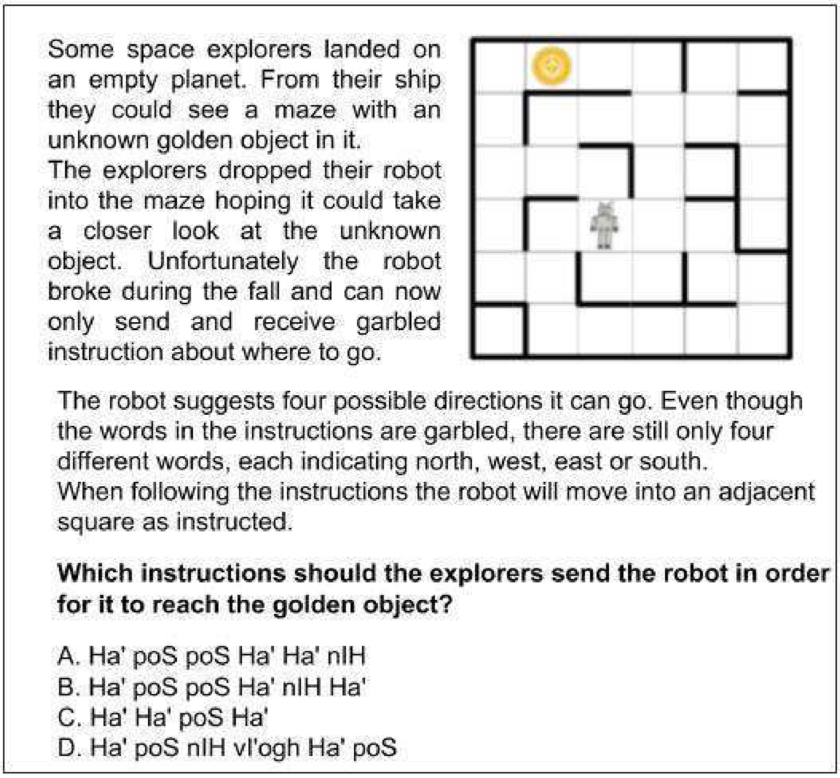


Figure 2.1: Space Maze task

2.3 Factor Analysis

Factor Analysis (FA) is a multivariate statistical technique which is suitable for examining the patterns of multidimensional relationships [41]. It allows us to present the structure and interrelations of a group of variables. This structure comes from the analysis of the underlying patterns for a set of variables. The interrelations can point out whether the information can be condensed in a group of factors. When the variables are overlap (i.e., correlating), then we can identify the factor.

Factors represent constructs, i.e., the underlying dimensions of a set of observed variables. In addition, variables can be items, questionnaire answers. Thus, one factor (construct) is related to variables by *factor loadings*. The factor loadings are the correlation between the variable and the factor, which are the measure used to understanding the essence of a factor. So, a factor model shows how well the variables are clustered into the factors often represented by a factor matrix.

In general, *exploratory factor analysis* and *confirmatory factor analysis* are the two main techniques of factor analysis. The exploratory analysis is a descriptive process intended to find a suitable factor model, whereas the confirmatory analysis is a hypothesis test-driven process to endorse a factor model. In this Section, we highlight the essential terms and procedures. More methodological details can be found in Hair et al. [41] and Brown et al. [15].

2.3.1 Exploratory Factor Analysis

In the exploratory factor analysis, we have not sure how the data behavior are established and how the variable are correlated. Then, we have to look into the data in order to condense the information in a smaller set of variable with a minimum lass of information. In other words, we try to figure out the factor model.

Variance is a principal concept when we are work with factor analysis. Variance is a value calculated from the square of the standard deviation. The meaning of variance is that, for a single variable, the total amount of dispersion about its mean represent the variance. Moreover, correlated variable shares variance. The amount of this variance is calculated form the squared correlation. Hair et al. illustrated that "if two variables have a correlation of 0.50, each variable shares 25 percent ($.50\hat{}2$) of its variance with the other variable." [41] (page 103). Thus, we understand what the meaning of variance is.

We summarize the PCA procedures in four-stage [41], as follows: *A - Checking the conceptual assumptions to employ factor analysis*; *B - Deriving factors and assessing the overall fit* ; *C - Rotate factors and (if needed) specify the Factor Model again.*; *D - Naming factors*

A - Checking the conceptual assumptions

Initially, we check some conceptual assumptions. Firstly, we look into the sample size. The minimum of the sample size is five subjects per question/variable, but it is preferably the sample size should be 100 or larger [41]. Secondly, we calculate the Measure of Sampling

Adequacy (MSA), which shows the appropriateness of carrying out factor analysis. This value ranges from 0 to 1, reaching 1 when the variables are perfectly predicted, and 0 as the opposite. The results should be above 0.50 for either the entire matrix or a single variable. Then, we check the Bartlett test of sphericity. This statistical test shows the presence of correlations among the variables and their significance. If the significance is < 0.05, it indicates sufficient correlations among the variables to proceed with factor analysis. Thus, these measures are accountable for quantifying the degree of intercorrelations among the variables. Therefore, they reveal the appropriateness of factor analysis.

B - Deriving factors

The exploratory phase can be conducted by Principal Components Analysis (PCA) or Common Factor Analysis. The difference is the extracted factors are based on total variance in PCA, while in Common Factor Analysis the factors are based only on the common variance [41]. Regardless the technique, the procedure are very similar. Here, we describe the PCA.

We explain this phase using a random example. Table 2.2 shows the PCA factor matrix. The first column is the variables (or the items in a questionnaire). The second to fourth are the results for the three factors that are extracted. In other words, the numbers in these columns represent the factor loadings of each variable on each of the factors. The factors are extracted in order, i.e., factor one accounts for the largest amount of variance, followed by factor two, at lastly, factor three. We highlighted the higher factor loadings in each factor, i.e., that loadings greater than 0.40 [41]. As we see, items 02, 04, 05, and 07 have loadings on factor one, while items 01, 03, and 06 on factor two. Particularly, factor three has only one high loading. A factor with a single variable (item) is not desirable because the interpretation would be difficult in the future and theoretically no meaningful. Therefore, factor Three should be removed from the factor model. Besides, item 7 has a cross-loading in factors One and Three. A *cross-loading* occurs when a variable has two more factor loadings exceeding the threshold value. Usually, the difference among loading should be greater than 0.20 [41]. If it was lower than 0.20, the variable should be drop out.

C - Rotate factors and redefine the Factor Model

After identifying the significant loadings in the unrotated factor matrix, we should try rotate factors in order to improve the interpretation of factors. The benefits of trying rotate

Table 2.2: Example of factor matrix

Variables	Factor One	Factor Two	Factor Three
Item 01	0.15	**0.89**	0.03
Item 02	**0.76**	0.11	0.12
Item 03	0.11	**0.69**	0.09
Item 04	**0.82**	0.05	0.10
Item 05	**0.69**	0.04	0.08
Item 06	0.19	**0.74**	0.21
Item 07	**0.59**	0.13	**0.67**

factor is, as Hair et al. said, "... to redistribute the variance from earlier factors to later ones to achieve a simpler, theoretically more meaningful factor pattern" [41] (page 111). Even with rotation, the total amount of variance extracted is the same comparing the rotated and unrotated solution. Thus, we should select the better factor modal, with or without rotated, based on the higher factor loadings.

Next step, we assess the values of factor loadings in order to select the most significant factor loadings. This will conduct to the the simplified structure as well as more representative. Whether any issues remain, like cross-loading more than 0.20, non significant loadings for one or more variables, or lower communalities, we must redefine the factor model. For example, we should withdraw variable by variable and calculate the factor loadings again until reach a appropriated factor model.

D - Naming factors

When a acceptable factor model has been derived, we attempt to nominate the factors. The process of naming the factors is theoretical and includes substantive interpretation of the factors intended to give singular meaning to each one. The variables with higher loadings usually influence more the label of factors.

The practice of naming factors is based essentially on the subjective view of the researcher. It is possible that another researcher find different name to the same factors considering her/his backgrounds and training in the area. In light of this aspect, the process of naming factors is critical and passive of questioning. The literature recommends that a strong theory should be present to justify the solution.

2.3.2 Confirmatory Factor Analysis

The Confirmatory Factor Analysis (CFA) is employed when we need to test a theory or a model into a data [41]. In the first case, the theory came from a well-know area by other theoretical studies. In the second, the model is the result of exploratory factor analysis, such as PCA. Regardless of the origin of the structure, the aim is established a reliable view of the factors by the goodness of fit.

There are particular measures which we calculated to determine the goodness of fit, known as fit indices [45]. The Comparative Fit Index (CFI) indicates if the model fits the data better than a more restricted model, whereas the Tucker-Lewis Index (TLI) penalizes over complex models. Both values ≥ 0.90 are acceptable, but higher is better. The Akaike's Information Criterion (AIC) and Schwarz's Bayesian Information Criterion (BIC) attempt to select models which show the most efficient representations of the observed data. They look for the most parsimonious representations, and lower values are better. RMSEA (Root Mean Square Error of Approximation) and standardized root mean square residual (SRNR) refer to residuals errors. As expected, lower values are better, but for the first, ≤ 0.1 is accepted, while ≤ 0.08 is admissible for the second.

In summary, the fit indices, as mentioned earlier, indicate whether the model should be accepted or denied. Moreover, the model must be supported by a theory which represents the underlying dimensions of the observed variable. The researchers have responsibility for the theory which sustains the model.

2.4 Network Analysis

In Network Science, a network can be understood as a visual representation of empirical data through mathematical models which combine algorithms and graphical techniques [47]. A network represents a system in which the variables or objects are *nodes*, also called *vertices*, and the relationships between these nodes are represented by lines or arrows, called *edges*. Networks are applied in Computer Science, Physics, Biological Sciences, Social Sciences, and Psychometric.

Networks can not only represent complex phenomena but also identify correlations between variables [30]. In the last case, it is usual to illustrate variables as nodes. Particularly, important relationships, as correlations, can be detected without the necessity for data reduction methods. In fact, the researcher can represent complex statistical patterns using networks.

A network has several properties which we highlight the weight and direction of a edge. Weight indicates the strength of the connection, and edges may or may not be directed [31]. In unweighted nets, the lines only indicate that there is a relationship between the variables, without showing the intensity. In the weighted networks, besides indicating the relations, the edges also indicate the direction of the association (employing colors) and the intensity of these relations (by the thickness). By default, green edges indicates a positive relationship and red edges indicates a negative relationship [30]. The color saturation and the width of the edges indicate the the absolute weight and scale relative to the strongest weight in the graph. Thicker edges indicate stronger relationships, while thin edges indicate weaker ones.

Under Psychometric, network analysis can be employed to identify information emerging from empirical data [47; 13; 31]. For instance, whether the data of an instrument are applied network analysis, we can group the items of the instrument and then discuss the latent variables. Therefore, network analysis is a forthcoming technique to identify constructs.

2.5 Item Response Theory

Psychometric theories have been applied in the measurement of latent variables [32]. Although a latent variable cannot be directly observed, its existence can be inferred from manifest variables. Item Response Theory (IRT) is a method for the design, analysis, and scoring of instruments which measure latent variable, such as abilities or attitudes [42].

IRT is a mathematical testing model which is based on an individual's performances on a test designed to measure specific abilities [6]. Baker explains that "the basic concepts of item response theory rest upon the individual items of a test rather than upon some aggregate of the item responses such as a test score" [6] (page 6). In other words, while the Classical Test Theory (CTT) is based on total score, under IRT, the baseline is whether the examinees respond to each item correctly or not, rather than overall score.

According to IRT, three criteria can be examined in an instrument [6; 7]. The first is **discrimination** of an item. Discrimination describes how well an item can differentiate students with higher and lower levels of knowledge. In general, items with higher discrimination detects a small change in the ability level of students. The second criterion is the **difficulty** of the item. The purpose of the difficulty is to put the item into position along the ability scale. For example, an easy item works among the low-ability examinees and a hard item works among the high-ability examinees. At last, in tests with multiple-choice questions, there is always the possibility of correctly guessing when giving a random answer. Thereby, the correct response includes a small probability of **hit due to guessing**.

The three-parameter logistic model (3PL) is used for analyzing discrimination, difficulty, and probability of hit due to guessing. The fundamental equation of the 3PL model (Equation 2.1) is the probability that a randomly selected examinees with proficiency Θ will correctly respond to item j, characterized in terms of a is the slope of parameter of item, describing its sensitivity to proficiency; b is the threshold parameter of item j, determining its difficulty and c is the lower asymptote parameter of item j, reflecting the chances of students with very low proficiency selecting the correct response (guessing) [6]. In the equation, e is the base of the natural logarithm that is a constant 2.718. The theoretical range of ability Θ scale is infinity, but frequently, the practical range is from -3 to +3. By the way, values beyond this range are possible [25].

$$P(\Theta) = c_j + (1 - c_j)\frac{1}{1 + e^{-a_j(\Theta - b_j)}} \tag{2.1}$$

There are the two-parameter logistic model (2PL) and the one-parameter logistic model (1PL). The 2PL represents the equation of item parameters b (difficulty) and a (discrimination), as shown in Equation 2.2 [6]. Therefore, 2PL defines the discrimination and difficulty level of item, excluding the guessing (c parameter).

$$P(\Theta) = \frac{1}{1 + e^{-a_j(\Theta - b_j)}} \tag{2.2}$$

Finally, the one-parameter logistic model (1PL) or Rasch Model characterizes only the difficulty level of items. Under the Rasch Model, the discrimination parameter is fixed at a value of $a = 1$ for all item. Thus, only the difficulty parameter b is estimated, as shown in Equation 2.3. As we can notice, the equations of 3PL, 2PL, and Rasch model are similar, but it depends on which parameters we want to estimate.

$$P(\Theta) = \frac{1}{1 + e^{-1(\Theta - b_j)}} \tag{2.3}$$

Item Curve Characteristic (ICC) allows the graphic visualization of difficulty and discrimination parameters. ICC is an increasing monotonic mathematical function which predicts the behavior of the individual in an item. The S-shaped curve describes the relationship between the probability of a correct response (axis Y) given to an item and the ability scale (axis X)[6].

Figure 2.2 provides a typical ICC with difficulty level $b = 0$ and discrimination $a = 1$. The discrimination parameter a reflects the steepness of the item and describes how many individuals of different abilities are distinguished as to the probability of hitting the item. Figure 2.3a shows two ICC with the same level of difficulty but different levels of discrimination. When discrimination is greater than moderate, the item characteristic curve is S-shaped and steeper slope in its middle section. On the other hand, when the item discrimination is less than moderate, the item characteristic curve is nearly linear and appears rather flat.

Figure 2.2 and 2.3b demonstrates the difficulty parameter which is measured in the same ability scale Θ when the probability of the correct answer is 0.5 (or 50%) [6]. In both case, $b = 0$. Figure 2.3b presents three ICC with the same discrimination but different levels of difficulty. When an item is easy, the difficulty parameter occurs at a low ability level (the more left curve). Meanwhile, when an item is hard, it corresponds to a high ability level (the righter curve).

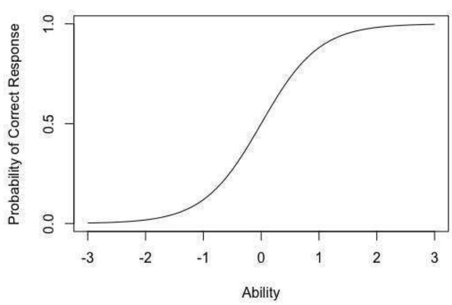
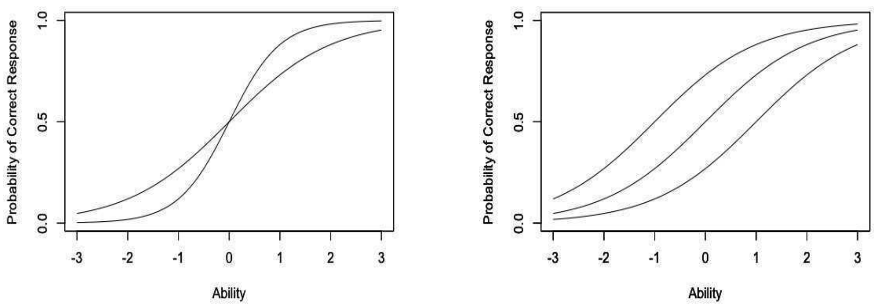


Figure 2.2: Typical Item Characteristic Curves

The discrimination parameter value a can be classified according to the power that an item has to differentiate examinees with higher and lower levels of ability. Usually, the



(a) Two ICC with the same difficulty level but different discrimination

(b) Three ICC with the same discrimination but different difficulty level

Figure 2.3: Item Characteristic Curves (ICC)

Table 2.3: Labels for item discrimination parameter values [Baker and Kim, 2017]

Label	**Range of values**	**Typical value**
None	0	0.00
Very low	.01 - .34	0.18
Low	.35 - .64	0.50
Moderate	.65 - 1.34	1.00
High	1.35 - 1.69	1.50
Very high	> 1.70	2.00
Perfect	+ infinity	+ infinity

probability of endorsing the correct answer increases as the ability level increases. Table 2.3 shows the range of values for seven labels based on [7]. For negative discriminate values, the probability of endorsing the correct answer decreases as the ability level increases.

When we observe the values of discrimination, difficulty, and item guessing parameter, it is important to notice critical values. The following problems can happen *i* the discrimination parameter is lower than 0.30, *ii* the difficulty parameter is lower than -2.95 or higher than 2.95, and *iii* the probability of hit by guessing is higher than 0.35 [7; **?**]. When these cases occur, the item should be examined and probably should drop out of the test.

Although the difficulty parameter has not a consolidate classification, we use the interpretation of Hambleton associate with Baker and Kim in this thesis [42; 7]. Considering that the item difficulty is a location parameter on the ability scale, Hambleton claims that item with $b < -2$ is a very easy item and with $b > +2$ is a very hard item. Baker and Kim suggest that if an item difficulty is -1, it works on lower-ability persons. On the order hands, for an item which difficulty is +1, it functions among higher-ability individuals. Then, we can conclude that between $b > -1$ and $b > +1$ is a medium range of difficulty. In addition, difficulty levels between $b > -2$ and $b < -1$ are easy and between $b < +1$ and $b > +2$ are hard.

2.6 Summary of Chapter

Although CT can be related to several definitions and skills, we initially adopted that CT is a problem-solving approach which is supported by cognitive abilities, such as abstraction, algorithmic thinking, decomposition, evaluation, and generalization. These skills are essential in CS and programming activities but can be stimulated without programming practices. Following this idea, Bebras Challenge is an international competition which promotes CT among students. The contest is composed of tasks (question) to all students ages which are answered without requiring CS or programming knowledge but stimulate CT skills.

The last part of the chapter briefly described the methodological concepts. Factor analysis is a statistical method for investigating interrelations of a group of variables, known as constructs or factors. Factors represent underlying dimensions of a set of observed variables. Exploratory factor analysis, as a Principal Component Analysis (PCA), is a descriptive process intended to identify an acceptable factor model, while the confirmatory analysis (CFA) is a hypothesis test-driven process to validate a factor model or test a theory. Both procedures are adopted to figure out or approve latent variables in instruments, respectively. Network analysis takes advantage of graphs and visual representation to represent latent variables under Psychometric research. Finally, IRT is a method intended to design and analyze instruments which is based on an individual's performances instead of the total score. The difficulty, discrimination, and hitting due to guessing are three parameters of analyze. The difficulty level is based on the examinees ability have a chance of 50% to endorse the correct answer, whereas discrimination is the item's capability of differing examinees' ability. At last, in a multiple-choice instrument, there is always a small probability of hitting due to guessing when the examinees give a random answer.

Chapter 3

Model and Procedures of Analyzing Computational Thinking Skills

This chapter presents the theoretical model of CT derived from the overall findings of our doctoral research. Then, we describe the general idea of providing the methodology strategies and instruments in order to quantify CT skills and how we have conducted the research toward this intended.

3.1 Theoretical Model of Computational Thinking

Recent studies recognize that CT is a development concept and should be an ongoing investigation [78]. Since Wing revisited the idea of CT in 2006, researchers around the world have been carried on the studies trying to make progress on the area [82; 14; 39; 50]. The fact is that we do not have a consensus of what CT means. We observed that some studies adopt previous models like those proposed by CSTA, ISTE, Computing at School as institutions intended to promote Computer Science on school or by literature review [8; 83; 18]. However, those previous models have two main limitations, which are the theoretical models based on literature reviews and the terminology.

Some of the main previous models are based on theoretical proposals or literature reviews published until 2015 and should be revisited in order to we have a consolidated definition [48]. The issue of theoretical proposals is the lack of robust empirical evidence or statistical procedures to support the allegations [55; 62]. Despite that, we recognize that tacit knowledge, as well as observation, are both important when we are investigating cognition abilities that are little known. However, only theoretical approaches are not enough to improve CT field.

Although the literature review is a successful methodological procedure in other Sciences, CT is in its infancy. The systematic literature review is a standardized analysis of published sources widely used in different fields like Medicine, Physics, and Social Studies. However, the key to success in these fields is they have many years in published research, and careful analysis can be done, as a meta-analysis. For example, some systematic review reports hundreds of studies [74]. Comparing to CT, we have a little more than one decade in publishing, considering the Wing's seminal paper in 2006. In other words, CT is just starting to be developed. Therefore, we are limited to what researchers published until that time and what frequency the studies are produced and accepted by the community. For example, a well-known model which inspired other models is proposed by Computer at School [18; 82], a UK community which the Bebras Challenge adopted [22]. Considering the growth of interest in CT, mainly in the last five years, the definition and associated abilities should be reviewed intended to evaluate if they are entirely understood and explored.

Another problem is that the terminology used in CT studies can be imprecise. Abstraction, evaluation, and generalization are examples of frequently used words. However, those terms not only have a double meaning but also can be interpreted differently depending on the study. For example, abstraction can be seen as being focusing on the essence of the problem [19] but also as selecting the representation of a system [22], building a model [8], or understanding the problem in different layers of details [18; 88], or finally, it can be abstraction of functionality and abstraction of data [82].

Instead of using words that can have several meaning or sub-abilities, it is advantageous we use the specific skills which we want to observe and measure. Firstly, using more precise definition facilitates the understanding of CT abilities by the CT community as well as teachers who are not a computer scientist [16; 89]. Secondly, words and expressions used in CS can frustrate some teachers because they can think that CT is more complicated to apply than it really is. The name of the skills like "algorithmic thinking" can be seen as jargon for non-CS professionals and intimidate teachers into being attracted by CT [35; 60]. Finally, the granularity of abilities can assist the development of new approaches to assess CT more properly. The level of detail present in describing the skills can make the CT more operational and measurable if we apply the suitable methodology.

Considering the context, we propose the theoretical model of CT which was direct influenced by the above mentioned problems and the results of our investigations. These studies include the analysis of how many skills CT encompasses through statistical procedures (Chapter 4), the evaluation of CT questions and which abilities are involved to solve them (Chapters 5 and 6), the state of art in CT [24](Section 2.1), and psychometric principals of cognitive assessment [33; 47; 7].

First of all, it is essential to define what type of problem we deal in CT. In this case, the CT problems should be *well-defined* and *knowledge-lean*. *Well-defined problems* means that the problem should have a precise specification to be solved, including the beginning, objective, and methods available for elucidating the solution [33]. An opposite view is the ill-defined problems, which have an inaccurate definition of the problem statement and how to solve it is usually unclear. An example of ill-defined problems is "how can we reach success?". Particularly, *well-defined problems* can be solved by an optimal strategy and always have a known right answer, differently from ill-defined problems. As a consequence, it is possible to identify the errors and inadequate strategies in processes to settle the problem [33]. Meanwhile, *knowledge-lean problems* denotes that the problem should not require ample amounts of prior knowledge to find the solution. In this kind of problem, the essential information to answer it is granted by the problem statement. Unlike *knowledge-lean* problems, knowledge-rich problems can be solved only if we have some knowledge on the subject [33]. Considering the different kinds of problems, it is important to highlight that we only treat *well-defined* and *knowledge-lean* problems in CT.

After claiming that CT problems are well-defined and knowledge-lean, we move forward to examine how the skills are covered. Significantly, the studies mention that the abilities are directly related to CT [8; 38; 24]. However, the attempt to see the skills as dimensions or factors of CT failed, as the statistical results showed in Chapter 4. Even though our findings have suggested that CT is a single dimension, the connection with operational skills should be rethought. Therefore, we saw the necessity to reorganize how the CT and abilities are connected to each other. We noticed that competences could be a link between CT and the abilities. Competence is the capability to coordinate prior knowledge or skills in order to deal with new situations or problems [69; 71]. In this direction, we have seen competence as proficiency in dealing with a set actions in order to reach an objective. In the same way, ability or skill is the faculty in executing an action intended to fulfill the goal. Indeed, competence is a high level of accomplishing the objective while the ability is the more specific (low) level of activity. The relation between competence and ability seems as complementing each other, but distinct levels of achievement. Then, CT (as a dimension or factor) can be related to one or more competence, which can be associated with one or more abilities/skills, as shown in Figure 3.1.

Figure 3.1: Relations among dimension (factor), competence, and ability/skill

Based on the results on Chapters 4, 5, and 6, we suggest that CT can have four main cognitive competencies, as follow: (i) understanding the problem, (ii) analyzing the data, (iii) thinking about instructions, and (iv) proposing a solution. Each of these competencies has some of associated skills which assist the student in settling the problem. Figure 3.2 summarizes the CT model. The associated skills are defined as operational as possible in order to facilitate the understanding. We believe that using the meaning of skills may help to set visibility of CT in research as well as in schools because teachers and other no-CS professionals can understand CT without computational words. This approach can also enable more interdisciplinary initiatives considering that teachers will have more independence in applying the CT principals in their activities if they master the CT concept and skills. As follows, we will describe the competencies and skills.

Understanding the problem competence is crucial in the real world when the problem appears to confuse or has data exceedingly. It is the domain of organizing the problem in a different way so that it is clearer or more strongly expressed to better understand [18]. Intended to make the problem statement understandable, we can use the following operational skills: *highlighting the essential data*, *recognizing sub-problems*, and *thinking like component parts*. *Highlighting the essential data* is the act of filtering out the key information in order to understand the essence of the problem and the data that will help to solve the

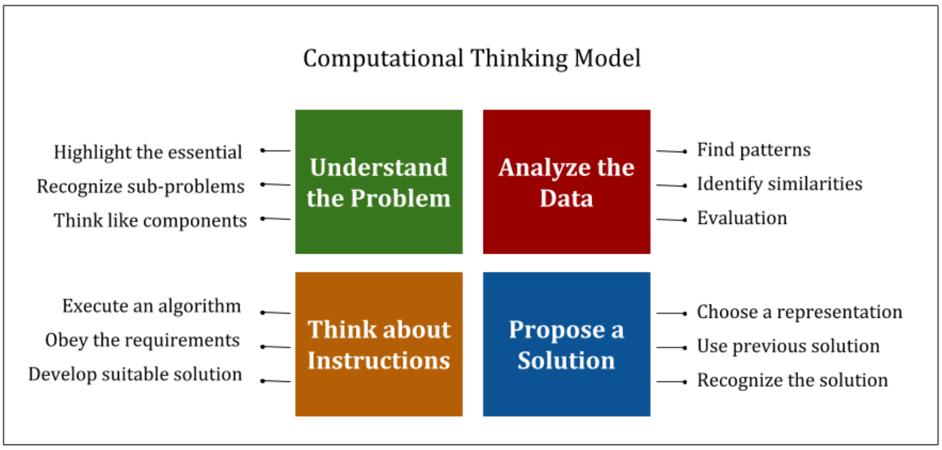


Figure 3.2: Computational Thinking Theoretical Model

problem [8; 18; 57]. It is the capability of withdrawing irrelevant details that will not affect the solution and maybe confuse the solver. *Recognizing sub-problems* means breaking down problems into smaller, manageable parts. Some studies call these skills decomposition [8; 18; 22]. Finally, *thinking like component parts* means considering the problem as elemental units more manageable that put together shape the problem [57; 20]. The two last skills can provide a new representation of the problem domain, creating a new perspective of the problem, making it easier to grasp [8]. All these abilities assist the solvers to reduce complexity, see the problem more clearly, and organize the next step of the solution.

Analyzing the data competence involves the strategies used to derive the solution by examining the available data in the problem statement. It includes making sense of data by analyzing and verifying predictions toward concluding the task [18]. The skills that can be involved are the following: *finding patterns*, *identifying similarities and connections*, and *evaluation*. *Finding patterns* means to identify the regular way in which some data or something happens or develops [18; 82]. It also includes recognizing strategies that are a non-trivial plan of acting to tackle the problem [57]. *Identifying similarities and connections* is important when we found something in common or understand how something makes sense together [22]. *Evaluation* is the ability to find the best solution to a problem considering the resources. It encompasses making judgments based on the requirements and context.

Thinking of instructions and circumstances means acting based on the requirements or making a judgment about a situation after considering the available information [57]. The skills that can be involved are the following: *executing an algorithm*, *obey-*

ing the requirements, and *developing suitable solution*. *Executing an algorithm* is the skill of producing a solution following a set of order steps. It also includes to comprehend some formal procedure or codification [8; 18; 22; 82; 57]. *Obeying the requirements* involves to think and reason as it was described precisely based on requirements and rules [22]. *developing suitable solution* encompasses to analyze whether the data have the right qualities or is fitness for a particular purpose based on the instructions or rules [22; 57].

Proposing a solution competence is the domain of organizing the data during the resolution process and getting directions in order to achieve the answer. Intended to produce a solution, we can use the following operational skills: *choosing a representation of a solution*, *using the previous solution*, and *recognizing a solution*. *Choosing a representation of a solution* encompasses to evaluate how to choose the satisfactory, effective, or proper response to the problem. It also includes to depict and organize the solution in a different representation, like graphs, words, images, or code. [22; 57]. *Using a previous solution* involves the act of solving new problems based on already-solved problems. It also encompasses to adapt the solution process to the new requirements [22]; *Recognizing a solution* is an essential ability when the problem has a compound response. It happens when the question has an open response, or the problem asks, for example, "which are the points eligible to solve the problem?" or "how many or which combination are possible considering the requirements?". In this case, the solver should notice that there is a combination of answers intended to the problem be solved entirely.

The CT model was designed after the overall investigation described in this thesis. In summary, we have analyzed the competencies and skills explored to solve two instruments which contain CT questions. The instruments (and the questions) will be detailed in Section 3.2. We have answered each question and contrasted the solution with those proposed by the organizers of the instruments. Then, we have identified the skills and related them to the literature definitions. During the process of analyzing the questions resolutions and the CT abilities proposed by organizers, we have noticed that this proposal has some limitations. We saw the skills from a different perspective, considering the actions to solve. So, we have chosen a proper name to distinguish each skill and competence. However, we have highlighted that these skills were not unprecedented; they are in accordance with state of the art in CT.

3.2 Roadmap of Studies

We carried out two main studies which are shown in Figure 3.3. In the first one, we examine the CT dimensions. In other words, we try to figure out how many and which factors encompass CT problems. Here, we consider factor as a latent variable. In the last one, we evaluate the CT items themselves in order to puzzle out what competences, skills, and contents are explored.

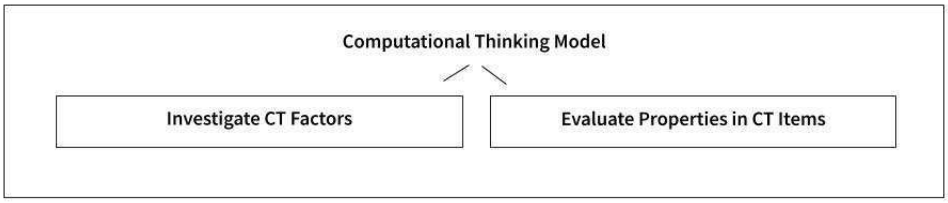


Figure 3.3: Overview of general procedures

3.2.1 Overview of Investigating Computational Thinking Factors

In our first study, we identify how many dimensions/factors are presented in CT problems. As a statistical procedure, we used factor analysis. We chosen FA because it provides statistical information about the relation of unobserved variables called factors [41]. FA is used in cognitive research by psychologists and social science researchers interested in several domains such as intelligence, attitudes, personality, beliefs [47; 32; 41]. This study is divided in two main parts, as shown in Figure 3.5. Initially, we verify the multidimensionality. However, the results showed that the CT instruments had not several dimensions. Then, we check the unidimensionality. In this last part, we found statistical support to claim that the used CT instruments were unidimensional, i.e., they measure only one factor (Computational Thinking itself).

Figure 3.5 details the first part of examining CT dimensions. In the beginning, we selected the instrument used in our study. Secondly, we organized the first data collection. Then, we conducted a Principal Component Analysis (PCA). The outcome was the exploratory model and the number of factors. So, we move on to confirm this model organizing

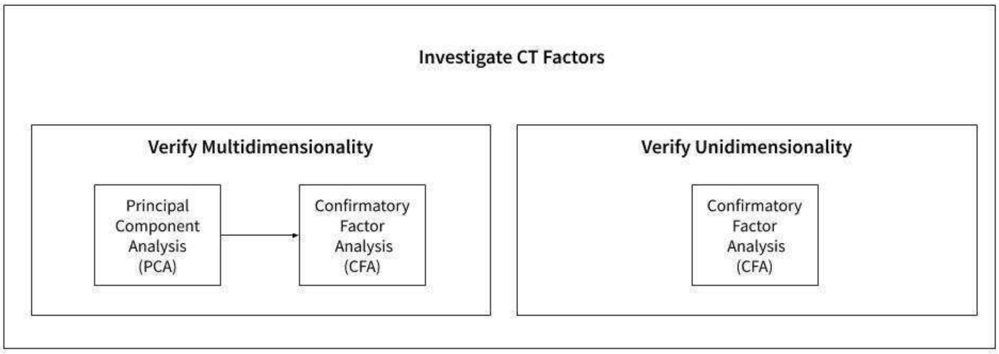


Figure 3.4: Overview procedures of investigating CT Factors

a new data collection. We carried out a Confirmatory Factor Analysis (CFA) and analyzed the statistical results to confirm or deny the model.

Later in this Chapter, we will apprise that we chose two instruments to conduct our study. Therefore, the steps described in Figure 3.5 were executed twice, one for each instrument. Nevertheless, the general finding was the same for both instruments.

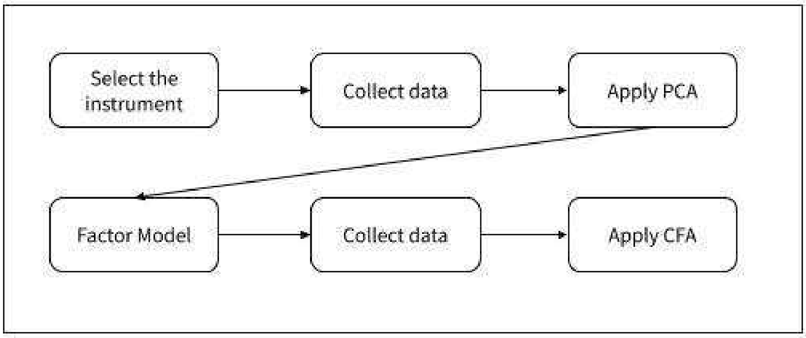


Figure 3.5: Details of multidimension analysis

The last part of this study verified the unidimensionality of the instruments. Thus, we conducted a CFA using the dataset of first data collection for both instruments considering one single factor. As a general result, we claimed that the instruments were unidimensional, i.e., they assessed the cognitive factor of Computational Thinking.

3.2.2 Overview of Evaluating Item Response Theory Properties in Computational Thinking Items

In order to discuss how CT items have been designed to evaluate CT skills, we adopted IRT. We chosen IRT as a methodological approach intended to take advantage of the benefits over the traditional methods, like classical test theory (CTT). Under CTT, the examinee's abilities are measured by the total score of correct answers in the test [25]. In this case, if two individuals have the same total score, they share the same ability level, even if the first one responded only the easy questions while the second one answered the medium and hard questions. In other words, the total score is not able to distinguish examinees who responded to questions with different difficulty levels [68]. On the other hand, under IRT, it is possible to detect the cognitive ability level based on some criteria, such as difficulty level [7]. Another advantage of IRT is the measure of the examinees' ability (Theta) is closer to reality because it does not depend on the sample, different from CTT.

In our study, we are concentrated on items' IRT properties instead of the ability level of individuals provided by the role instrument [7; 6; 68; 67; 47]. This approach was adopted because we did not have a reliable instrument to develop a thrust scale of CT ability at the moment. As an alternative, in order to design a reliable test in the future, we have focus on the analysis of items in this research. We have explore the difficulty and discrimination level in the items intended to learn the best qualities and characteristics. As Figure 3.6 shows, we have analyzed the items following the criteria of difficulty and discrimination levels provided by IRT.

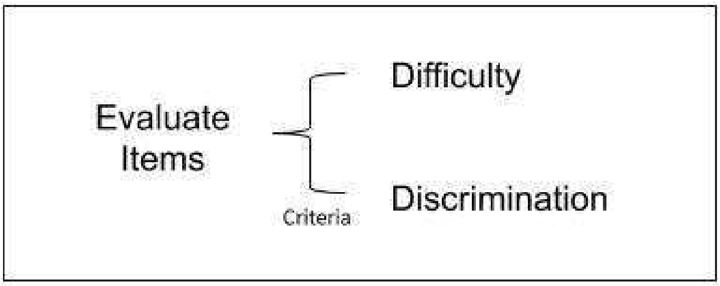


Figure 3.6: Overview of Evaluating items

Although the difficulty level of the items in instruments is established and discussed, it is often assigned by its organizers based on their subjective judgment, tacit knowledge, and experience as teachers, like in Bebras Challenge [22; 85; 58]. Therefore, it is important to use empirical data in order to provide the classification of difficulty level close to what it is measured.

On the other hand, discrimination of items is usually not addressed in studies of measuring CT skills. However, we claim that it is essential to know which items are capable of distinguishing examinees from those not valuable to assess students [7; 6; 68; 67; 47].

3.3 Ethical Principles

The Ethics Committee of the Federal University of Campina Grande approved this research project in August 2016. The study was registered at Plataforma Brasil[1] by the number CAAE: 56847316.7.0000.5182. The Ethics Committee judged that the project is in accordance with the Resolution number 466/2012 concerning the ethical principles.

All study subjects or their legal representative signed informed consent forms before the beginning of the study. The *"Termo de Consentimento Livre e Esclarecido"* is available in Appendix A (in Portuguese). The data were made anonymous with the intention to preserve the subjects' identity. Thus, it is not possible to identify the examinees.

3.4 Material and Methods

In this section, we present the instruments, the participants, and the adopted procedures in our study.

3.4.1 Instruments

Bebras Challenge was the chosen instrument to be investigated in our study for many reasons. Firstly, the tasks are designed to be answered by exploiting CT abilities requiring no prior knowledge in CS or programming. Therefore, all students can answer the tasks without knowing programming or having been exposed to CS classes. Moreover, the tasks are

[1] More details about the research project we have submitted to the Ethics Committee can be found at http://plataformabrasil.saude.gov.br/ using CAAE: 56847316.7.0000.5182

independent, self-contained, and self-explanatory. Secondly, the tasks are not only designed by researchers around the world but also improved at an annual workshop by specialists. This process can help to produce better tasks since several international professionals have analyzed the task before the final version. Finally, the organizers in some countries produce the brochures of their national contest. These brochures contain the tasks of all ages and the respective solution. Therefore, other researchers can use those tasks for various research goals.

Among all brochures, we selected the brochure of national Bebras Challenge from the United Kingdom (UK) in 2015 [11] (available in Appendix D) and 2014 [43] (available in Appendix E). We had to choose brochures from a foreign country because, until 2018, Brazil has not yet participated in the Bebras Challenge. Consequently, there were no tasks available in Portuguese. Brochures from the UK were chosen because it was the only available contest in English at the beginning of our investigation. All other published brochures were available on several, less accessible, languages, such as Lithuanian, German, Swiss, Iranian. We have judged that English was not only our domain language but also we would find researchers who could assist us with the translation into Portuguese.

Considering that tasks have been designed for all age groups of students, we selected those tasks for the older group (over 16 years old) in accordance with the age of the participants in our study. The instrument One was the contest published in 2015 while instrument Two was that published in 2014. The name of the tasks is presented in Table 3.1 and Table 3.2. After choosing the instruments, we translate the tasks into Portuguese. Avoiding bias and translation mistakes, each task was reviewed by two external researchers from our study. Thus, the instrument One is shown in Appendix B whereas instrument Two is presented in Appendix C, both in Portuguese.

We should explain that instrument One has minimal difference considering the original brochures from the UK. Five tasks were replaced to three new tasks which came from Lithuania Bebras Challenge. This occurred in an attempt to contrast the results from Brazilian students to Lithuanian students. We have reported this study in Araujo et al, 2018 [2].*Turn the cards*, *Decorating chocolate*, and *Busy beaver* tasks were shared in English by Lithuanian organizers and translated into Portuguese. Unfortunately, our efforts to contrast the results were unsuccessfully (See paper [Araujo et al. 2018] [2]). Nevertheless, we have chosen

Table 3.1: Tasks of instrument One

Item	tasks	Item	tasks
i01	Drawing stars	i07	Word chain
i02	Bowl Factory	i08	Fireworks
i03	Email	i09	You won't find it
i04	Beaver the alchemist	i10	Turn the cards
i05	Tutorial	i11	Decorating chocolate
i06	Popularity	i12	Busy beaver

Table 3.2: Tasks of instrument Two

Item	Tasks	Item	Tasks
i01	Ceremony	i09	Social network
i02	Log-art	i10	Height game
i03	Beavers on the run	i11	Meeting point
i04	Traffic in the city	i12	Best translation
i05	Storm proof network	i13	Broken machines
i06	Space maze	i14	True or false
i07	Footprints	i15	Right rectangles
i08	Puddle jumping		

to continue our investigation using the tasks presented in Table 3.1. Finally, we have highlighted that instrument Two is the original made available by the UK organizers.

3.4.2 Participants

The participants of our study were recruited from an introductory programming course at Federal University of Campina Grande (UFCG) and Federal University of Paraíba (UFPB) in 2017 and 2018. In UFCG, there is one undergraduate program, the Bachelor in Computer Science, while in UFPB, there are two, Bachelor of Information Systems and Licentiate of Computer Science. Regardless of the undergraduate program, introductory programming course aiming to develop the students' programming skills using Python programming language in both universities.

We conducted two distinct data collection for each instrument (four main data collection in total). Thus, the number of participants is different in all situation. We will detail the sample size and the instrument in each circumstance.

In the first data collection of **instrument One**, the total sample consisted of 170 students, whose 150 are male and 20 are female while in the second data collection, the whole samples were 97 students, whose 87 are male and 10 are female. Considering the sample of the two universities, Table 3.3 shows the number of total participants (*n*) and participants per sex in two data collection of both universities. We emphasize that in the second data collection, only students from UFCG participated in 2018.2 (September 2018) because the semester in UFPB had not yet started.

Table 3.3: Participants of Instrument One by Universities

Data Collection	University	*n*	Male	Female
Firts	UFCG	98	86	12
	UFPB	72	64	8
Second	UFCG	97	87	10

In the first data collection of **instrument Two**, the total number of participants was 214, 189 of which are male, while 25 are female, whereas in the second data collection was 153, whose 137 are male and 16 are female. Considering the students of UFCG and UFPB, table 3.4 shows the sample size of each university. As we can see in the first data collection, the participant numbers differ because in the moment of data collection at UFPB, several students did not appear on the test day due to unforeseen circumstances. On the other hand, in the second data collection, the amount of students is very close.

Table 3.4: Participants of instrument Two by Universities

Data Collection	University	*n*	Male	Female
Firts	UFCG	188	165	23
	UFPB	26	24	2
Second	UFCG	79	72	7
	UFPB	74	65	9

3.4.3 Procedures of data collection

The instruments were organized in a printed version so that students could answer the questions using paper and pencil. We collected data from 2017 to 2018 at UFCG and UFPB.

The instrument One was applied at the begging of the 2017.1 and 2018.1 semesters while instrument Two was employed at the end of the same semesters.

The data collections were carried on by the main researcher or professors of the courses. In each procedure, we explained the objective of the research and ethical issues. All students or their legal representative signed the consent form before the beginning of the test. Table 3.5 illustrates the number of tasks for each situation. In first data collections, the students answered the tests in until 60 minutes, while in the second data collection, the students responded in until 40 minutes. The time was based on the Bebras Challenge instructions.

Table 3.5: Number of Tasks per Instrument and Data Collection

	First Data Collection	**Second Data Collection**
Instrument One	12 tasks	7 tasks
Instrument Two	15 tasks	8 tasks

After applying the instrument, the data was organized in a table. We scored one (+1) for each correct answer and zero (0) for an incorrect answer in both instruments. Next, we calculated the total score for each student, considering the total amount of correct answers. We emphasize that we do not adopt the point counting system of Bebras Challenge because our analysis is based on dichotomies items, i.e., correct or incorrect answers.

3.5 Summary of Chapter

In this research, we have defined that CT is an approach to solve well-defined and knowledge-lean problems supported by competencies and skills. Well-defined problems can be solved by an optimal strategy and always have a known right answer, while knowledge-lean problems do not require prior knowledge to find the solution. Regarding the competencies and skills, we have recognized that competence is a high level of cognitive ability to accomplish the goal, whereas the skill involves a specific level of actions. Then, we have proposed that CT encompasses four main cognitive competences. Each one of these competencies is associated with three main skills, which are summarized below.

- **Understanding the problem competence** involves to organize the problem indented to better comprehend what is to be solved. The associated skills include *highlighting* *the essential data*, *recognizing sub-problems*, and *thinking like component parts*.

- **Analyzing the data competence** encompasses to plain how to find the solution by examining the problem statement. The associated abilities comprise *finding patterns*, *identifying similarities and connections*, and *evaluation*.

- **Thinking about instructions competence** includes to act based on the requirements or make a judgment about a situation after considering the available information. The associated skills encompass *executing an algorithm*, *obeying the requirements*, and *developing suitable solution*.

- **Proposing a solution competence** involves to systematize the data during the resolution process and get directions in order to achieve the answer. The associated skills include *choosing a representation of a solution*, *using a previous solution*, and *recognizing the solution*.

Then, we briefly have described how factor analysis and item response theory were used as a statistical resource to conduct our research. We have emphasized that the Ethics Committee of the UFCG approved our research project. Also, we have detailed both adopted Bebras instruments. Then, we have specified the demographics, which have involved students from two universities. Finally, we have presented the number of questions in each case and how the data was collected and treated.

Chapter 4

Investigating Computational Thinking Factors

One of our objectives is to quantify CT skills reliably. For this propose, we need methodological support to sustain the necessity of a new theoretical model of competences and abilities. In this direction, we examine the granularity of CT like a latent trait, considering that skills can be factors.

In this chapter, we investigate how many factors of CT is represented by the selected instruments. Initially, we detail the data screening of all data collection. Then, we present the statistical procedures to support the allegation of this Chapter. Next, we demonstrate the results of verifying multidimensionality and unidimensionality. Finally, we discuss the reasons why we can not confirm the multidimensionality and why unidimensionality seems to fit better, considering the adopted instruments.

4.1 Data screening

In this section, we present the data screening concerning both instruments. For each instrument, we conducted two data collection. The first one was conducted using the complete instrument presented in Section 3.4: twelve tasks for instrument One (see Table 3.1) and fifteen tasks for instrument Two (see Table 3.2). The second data collection has a reduced number of items: seven tasks for instrument One and eight tasks for instrument Two but it is the same tasks. The explanation for the smaller number of tasks will be given through this

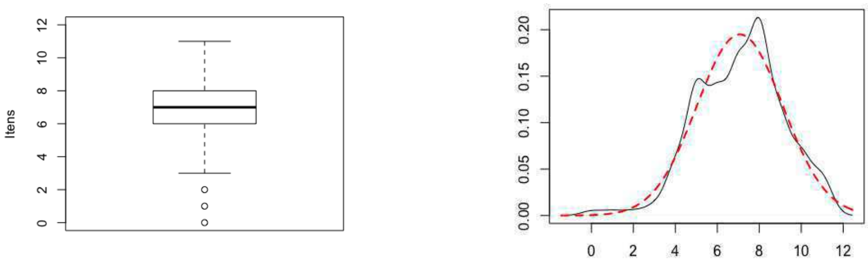
Chapter.

Apart from some outliers, all participants were aged between 17 and 21 at the moment of data collection. The numbers of male and female students are shown in each data collection. Due to the small number of female students in our data collection, we do not conduct any analysis of gender issues. Besides, when the data come from different groups, in our case, different universities, we analyzed whether the students came from the same distribution. This procedure is needed to ensure the students' answers in these two groups can be analyzed in the same dataset (together). For this purpose, we used the Chi-squared two sample test with a significance level of 0.05. The null hypothesis is that the two samples come from a common distribution, while the alternative hypothesis is that the two samples do not come from a common distribution. We also investigate the homogeneity of variance and the mean of scores from both universities using Levene's test, and t-test and Wilcoxon test, respectively. When the number of samples was different, we performed bootstrapping. The significance level of 0.05 was adopted to all hypothesis test performed in this Section.

4.1.1 Instrument One

In the first data collection of instrument One, the sample consisted of 170 students, 150 of which are male and 20 are female. We calculated the total score for each university group and carried on statistical procedures. Although both T-test and Wilcoxon test indicated that the mean score differ, the two samples have homoscedasticity, i.e., the score distribution of both universities presented homogeneity of variance. When we applied the Chi-squared two-sample test, the result indicates that two samples come from a common distribution. Therefore, we can analyze the samples together. Figure 4.1a shows the boxplot of total scores. Apart from some outliers, no one answered all twelve tasks correctly, and one student had any correct answer. The mean score was 7.08, and the standard deviation was 2.04. Although we can not affirm that the score follows a normal distribution (the p-value of Shapiro-Wilk test was 0.0003085), Figure 4.1b presents the density of score and the expected normal distribution in red dot line.

In the second data collection of instrument One, 97 students answered the seven tasks. This time, only students of UFCG participated in 2018.2 (September 2018) because the semester in UFPB has not been started. Figure 4.2a presents the boxplot of scores. We

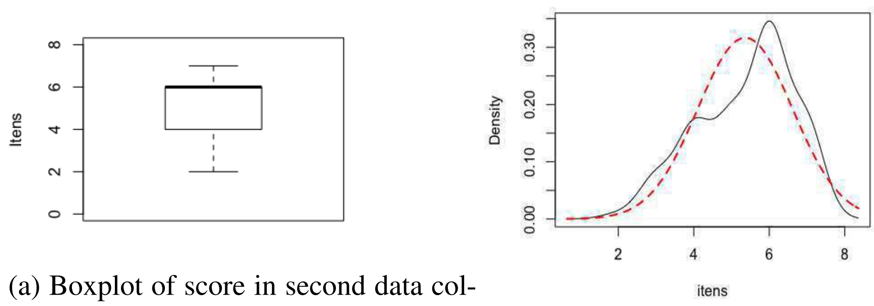


(a) Boxplot of scores in first data collection (instrument One)

(b) Density of score in first data collection (instrument One)

Figure 4.1: Graphics of Instrument One (first data collection)

notice that there are no outliers, and the lower score was 2 points. The mean score was 5.36 and the standard deviation was 1.25. Figure 4.2b shows the density of scores. The Shapiro-Wilk test indicates that the scores do not follow the normal distribution (the p-value of the Shapiro-Wilk test was 1.43^{e-3}).



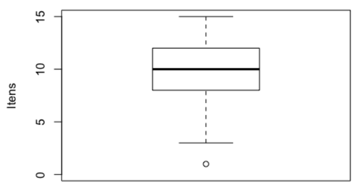
(a) Boxplot of score in second data collection (instrument One)

(b) Density of score in second data collection (instrument One)

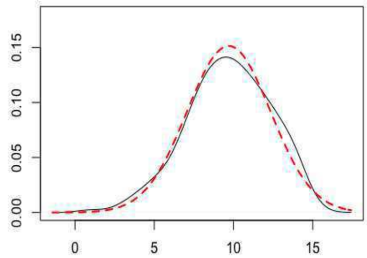
Figure 4.2: Graphics of Instrument One (second data collection)

4.1.2 Instrument Two

In the first data collection of instrument Two, 214 students answered the fifteen tasks. The boxplot in Figure 4.3a displays the distribution of data. The mean score was 9.701 and the standard deviation was 2.63. Figure 4.3b presents the density of scores. We notice that it is close to the normal distribution but the p-value of Shapiro-Wilk test was 0.001034.



(a) Boxplot of score in first data collection (instrument Two)



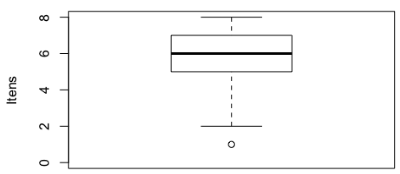
(b) Density of score in first data collection (instrument Two)

Figure 4.3: Graphics of instrument Two (first data collection)

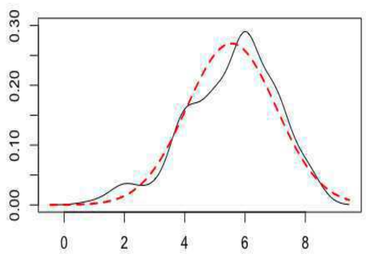
In the second data collection of instrument Two, the sample consists of 153 students, 137 of which are male and 16 are female. We also calculated the total score of each university and carried on statistical procedures. In summary, the means of the score are different based on T-test and Wilcoxon test (both p-values < 0.003), although the score variation has homoscedasticity and came from the same distribution. Therefore, the data can be analyzed together. Figure 4.4a displays the boxplot of total scores. We highlight that only eight tasks are presented at this time. Apart from some outliers, the mean was 5.549 and the standard deviation was 1.48. Considering the data distribution, the Shapiro-Wilk test indicates that the score does not follow the normal distribution (p-value $= 1,22^{e-3}$). Figure 4.4b shows the density and the expected normal curve in red line.

4.2 Statistical Procedures of Factor Analysis

All statistical analysis used in factor analysis were performed using the R language [72]. We conducted the PCA using psych library and *principal( ) function* with oblimin rotation [73]. Indeed, we used other measures. The Bartlett test should present p-values $\leq$ 0.005. The Kaiser-Meyer-Olkin (KMO) should show values higher than 0.8, but 0.6 is also acceptable. The Measure of Sampling Adequacy (MSA) is a measure of sufficiency to run CPA which



(a) Boxplot of score in second data collection (instrument Two)



(b) Density of score in second data collection (instrument Two)

Figure 4.4: Graphics of Instrument Two (second data collection)

values higher than 0.50 are acceptable. The eigenvalues were the chosen approach to estimate the number of factors in PCA [41]. In this case, we used *scree( ) function* and values ≥ 1. Likewise, we used factor loading higher than 0.3 and cross-loading lower then 0.2 to choose the appropriate number of factors [41].

For CFA, we used *lavaan* library and *cfa( ) function* [79]. As a measure of the goodness of fit, we used TLI (Tucker-Lewis Index), CFI (Comparative Fit Index), which values ≥ 0.90 is acceptable for both. We also used the Standardized Root Mean Square Residual (SRNR) considering ≤ 0.08 as allowable. Lastly, the Root Mean Square Error of Approximation (RMSEA) was considered admissible when ≤ 0.1. Finally, we adopted the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) as criteria in model selection, which lower values indicate better models [15]. These indexes were explained in Chapter 2.

4.3 Verifying multidimensionality

In this Section, we present the finding on verifying multidimensionality of instrument One and Two. For both instruments, firstly, we conducted a exploratory phase trying to find an appropriate number of factors derived from the first data collection for each instrument. The outcome of this phase was an exploratory model, one for each instrument. Then, we conducted a second data collection in order to confirm the exploratory model individually.

4.3.1 Multidimensionality of Instrument One

We initiated the exploratory analysis with PCA, calculating the goodness of fit to figure out whether we could continue with the factor analysis. As a result, all values were suitable for applying factor analysis [41]: (i) the Bartlett test indicated that there was a significant relation among the variables (p-values $< 2.2^{e-16}$), (ii) the KMO was 0.82, and (iii) all tasks presented MSA higher than 0.50.

The process of extraction of the final pattern matrix to build our exploratory model is described below. The eigenvalues point out four factors to be extracted (considering eigenvalues ≥ 1), but this did not arise from the data. When four factors were considered, only one task was loaded onto third and fourth factors, respectively. So, the analysis was reduced to two factors. Then, we calculated the pattern matrix with oblimin rotation because it produced the maximum values of loading and few cross-loadings. The PCA was initiated with twelve tasks. In the first iteration, "Bowl Factore", "Email", and "Beaver The Alchemist" tasks had cross-loading and were removed. "Turn the cards" task presented loading < 0.30 and was discarded (we considered 0.30 as a sufficient factor loading [41]). In the second iteration, "Fireworks" task presents MSA < 0.50 and was removed. After this time, all factor loading were > 0.30 in only one factor, KMO was > 0.76, and all MSA were > 0.50.

Table 4.1 represents the exploratory model of instrument One, which is the pattern matrix with two factors with oblimin rotation. Factor 1 was associated with "Drawing stars", "Tutorial", "Word chains", and "You won't find it" tasks, while Factor 2 was associated with "Popularity", "Decorating Chocolate", and "Busy beaver" tasks. Therefore, we will try to confirm this model with CFA.

Table 4.1: Pattern Matrix of Instrument One

Task	**Factor 1**	**Factor 2**	**Commun.**
Drawing starts	**0.47**	0.29	0.34
Tutorial	**0.67**	-0.04	0.45
Word chains	**0.71**	-0.11	0.5
You won't find it	**0.75**	0.05	0.57
Popularity	0.17	**0.59**	0.4
Decorating chocolate	0	**0.78**	0.6
Busy beaver	-0.14	**0.66**	0.44

The exploratory model was defined based on empirical data using the first data collection, now we move on to confirm it. We organized a new collected data in the same context of the first one. In this phase, the instrument only has seven tasks, which were described in Table 4.1.

In Table 4.2, we showed the factor loadings and standard error (SE) of CFA. The model fit presented good values. The Comparative Fit Index (CFI) was 1.539 and the Tucker-Lewis Index (TLI) was 0.99 (≥ 0.90 is admissible for both). The root mean square error of approximation (RMSEA) was 0.004 (≤ 0.1 is acceptable), while the standardized root mean square residual (SRMR) was 0.057 (≤ 0.08 is satisfactory). However, the standardized coefficient ranging from -0.241 to -0.010. Based on the negative factor loading, we have no statistical support to accept the model, despite it had good indicator of goodness of fit.

Table 4.2: Confirmatory Factor Analysis of Exploratory Model One

Latent Factor	**Task**	**Factor Loading (SE)**
Factor 1	Drawing starts	-0.077 (0.063)
	Tutorial	-0.056 (0.049)
	Word chains	-0.021 (0.018)
	You won't find it	-0.010 (0.020)
Factor 2	Popularity	-0.166 (0.064)
	Decorating chocolate	-0.241 (0.077)
	Busy beaver	-0.168 (0.070)

4.3.2 Multidimensionality of Instrument Two

Similar to section 4.3.1, we carried out an exploratory analysis calculating the goodness of fit of instrument two in the first data collection. In summary, PCA can be conducted because all values are acceptable: (i) the Bartlett test indicated that there was a significant relation among the variables (p-values $< 2.2^{e-16}$), (ii) the KMO was approximately 0.70, and (iii) all tasks presented MSA higher than 0.50.

The extraction of the exploratory model was conducted as follow. The eigenvalues indicate five factors (eigenvalues ≤ 1) but this number of factors showed five cross-loading and only one task had loading onto the fifth factor. Then, we reduced our analysis for the appropriate number of three principal factors. Next, we calculated the pattern matrix with oblimin rotation for the reason that it produced the highest values of loadings. The PCA begun with fifteen tasks. When we run the statistical procedures at the first time, "Ceremony" task had factor loading lower than the critical value (≤ 0.3). Meanwhile, "Traffic in the city", "Storm proof network", "Puddle jumping", "Best translation", "Height game", and "Meeting point" tasks presented cross-loading (≥ 0.2). All the above-mentioned tasks were removed from the analysis.

Table 4.3 presents the exploratory model of instrument Two, which shows the pattern matrix with three factors with oblimin rotation. Factor 1 was associated with "Log-art", "Beavers on the run", "Space Maze" tasks, while Factor 2 was related on "Broken machines", "Right Rectangles", "True or False" tasks. Finally, Factor 3 was associated with "Footprints", and "Social network" tasks.

Table 4.3: Pattern Matrix of Instrument Two

Task	**Factor 1**	**Factor 2**	**Factor 3**	**Commun.**
Log-art	**0.61**	0.02	-0.16	0.37
Beavers on the run	**0.73**	-0.16	0.16	0.59
Space Maze	**0.78**	0.13	-0.04	0.63
Footprints	0.03	**0.82**	-0.01	0.67
Social network	-0.02	**0.7**	0.07	0.5
Broken machines	-0.01	0.07	**0.62**	0.4
Right Rectangles	0	-0.14	**0.72**	0.52
True or False	0.03	0.19	**0.67**	0.51

Following the factor analysis procedures, we move on to confirm the exploratory model. We conducted a new data collection containing the eight tasks emerged from the PCA. Then, we calculated the model fit. The Comparative Fit Index (CFI) was 0.882 and the Tucker-Lewis Index (TLI) 0.806. Both values are lower than the admissible value (≥ 0.90). The root mean square error of approximation (RMSEA) was 0.054 (≤ 0.1 is satisfactory). Meanwhile, the standardized root is mean square residual (SRMR) was 0.060 (≤ 0.08 is acceptable). In Table 4.4, we describe the factor loading and standard errors (SE) of CFA. As we can see, "Beaver on the run", "Space Maze", "Social network", and "Broken machines" tasks presented factor loading lower than 0.1 (one task in each latent factor). Therefore, despite some good values of index fit, the model can not be statistical confirmed.

Table 4.4: Confirmatory Factor Analysis of Exploratory Model Two

Latent Factor	Task	Factor Loading (SE)
Factor 1	Beavers on the run	0.097 (0.017)
	Log-art	0.155 (0.035)
	Space Maze	0.081 (0.027)
Factor 2	Footprints	0.367 (0.322)
	Social network	0.064 (0.069)
Factor 3	Broken machines	0.064 (0.050)
	Right Rectangles	0.121 (0.052)
	True or False	0.376 (0.118)

4.4 Verifying unidimensionality

In this section, we have demonstrated the unidimensionality of the instruments, which means that only one underlying trait is being measured by a set of tasks. Although Bebras assess several skills, unidimensionality would suggest that they are all related to the single latent trait of Computational Thinking.

In order to test whether the instruments are unidimensional, we conducted a CFA considering one single factor. CFA is adopted when we need a confirmation of numbers of factors. We used the first data collection as a dataset for both instruments because we have aimed to evaluate all tasks.

4.4.1 Unidimensionality of Instrument One

We have started the process obtaining the model fit of instrument One. We have carried on a CFA and calculated the index. The comparative fit index (CFI) was 0.825 and the Tucker-Lewis index (TLI) was 0.786. Both values are a little bit below the limit but we have continued the analysis. The root mean square error of approximation (RMSEA) was 0.048 (≤ 0.1 is allowable). Meanwhile, the standardized root mean square residual (SRMR) was 0.069 (≤ 0.08 is acceptable).

Table 4.5 shows the factor loadings and standard errors (SE) results of CFA. We have recognized that some tasks have factor loadings lower than 0.1 (close to zero). Observing other criteria, the Akaike information criterion (AIC) was 1529.492 , while the Bayesian information criterion (BIC) was 1601.425. These two indexes will be a criterion to make a decision in which model we will choose when we have removed the tasks with factor loading lower than 0.1.

Table 4.5: Confirmatory Factor Analysis of Instrument One

Tasks	Factor Loadings (SE)	Tasks	Factor Loadings (SE)
Drawing stars	0.181 (0.043)	Word chain	0.068 (0.016)
Bowl Factory	0.130 (0.043)	Fireworks	0.056 (0.054)
Email	0.069 (0.033)	You won't find it	0.156 (0.030)
Beaver the alchemist	0.279 (0.046)	Turn the cards	0.026 (0.028)
Tutorial	0.145 (0.032)	Decorating chocolate	0.178 (0.048)
Popularity	0.168 (0.048)	Busy beaver	0.071 (0.047)

After we have put these tasks (with factor loading ≤ 0.1) out of the analysis, we calculated the fit model again. Now, the Comparative Fit Index (CFI) was 0.900, and Tucker-Lewis Index (TLI) was 0.801 which is acceptable for the first and approximately admissible for the second, but not critical. The root mean square error of approximation (RMSEA) was 0.071 (≤ 0.1 is satisfactory) and the standardized root mean square residual (SRMR) was 0.062 (≤ 0.08 is acceptable). When we have contrasted the AIC and BIC values with those mentioned above, both indexes are lower now (AIC was 1084.680 and BIC was 1126.641). This result indicates parsimony. Table 4.6 shows the new factor loading. As we can see, the values are similar to those shown in Table 4.5 but without tasks with values lower than 0.1.

Considering all tasks in instrument One, based on factor analysis, these seven tasks in Table 4.6 have worked better together than those twelve tasks shown in 4.5. In Chapter 6, we will discuss that these tasks, which were excluded, have presented critical issues related to difficulty and discrimination level.

Table 4.6: Confirmatory Factor Analysis of Instrument One with Seven Tasks

Tasks	Factor Loadings (SE)	Tasks	Factor Loadings (SE)
Drawing stars	0.185 (0.044)	Popularity	0.176 (0.049)
Bowl Factory	0.154 (0.044)	You won't find it	0.130 (0.031)
Beaver the alchemist	0.295 (0.049)	Decorating chocolate	0.209 (0.049)
Tutorial	0.124 (0.033)		

4.4.2 Unidimensionality of Instrument Two

We have initiated the process calculating the model fit of instrument Two. The comparative fit index (CFI) was 0.787 and the Tucker-Lewis index (TLI) was 0.751. In this case, both values are lower then the accepted measures. The root mean square error of approximation (RMSEA) was 0.049 (≤ 0.1 is admissible), while the standardized root mean square residual SRNR was 0.064 (≤ 0.08 is allowable). Finally, the Akaike information criterion (AIC) was 3070.999 whereas the Bayesian information criterion (BIC) was 3170.981. These last two indicators will be useful as a criteria in model selection (lower values indicate better models).

Table 4.7 shows the factor loadings and standard errors (SE) of Instrument Two considering all tasks. We noticed that the tasks had weak to moderate direct effects from the factor. Particularly, "Ceremony", "Log-art", "Beavers on the run", and "Space maze" had poor factor loading (< 0.1). Apparently, these particular tasks fit poorly and should be remove from our analysis.

Table 4.7: Confirmatory Factor Analysis of Instrument Two

Task	**Factor Loading (SE)**	**Task**	**Factor Loading (SE)**
Ceremony	0.033 (0.034)	Social network	0.169 (0.040)
Log-art	0.098 (0.028)	Height game	0.176 (0.040)
Beavers on the run	0.069 (0.015)	Meeting point	0.134 (0.041)
Traffic in the city	0.208 (0.035)	Best translation	0.155 (0.040)
Storm proof network	0.198 (0.040)	Broken machines	0.142 (0.041)
Space maze	0.099 (0.020)	True or false	0.158 (0.038)
Footprints	0.143 (0.032)	Right rectangles	0.107 (0.040)
Puddle jumping	0.194 (0.027)		

Then, we have recalculated the model fit of instrument Two withdrawing the abovementioned tasks (with factor loading ≤ 0.1). Now, the model fit has appeared better. CFI was 0.908 and TLI was 0.900, which are both acceptable now. Meanwhile, RMSEA was 0.038 (≤ 0.1 is adequate), whereas SRNR was 0.056 (≤ 0.08 is allowable). Both values are lower and better than the first model. Lastly, AIC was 2832.495 and BIC was 2905.815. Therefore, considering that not only all values are acceptable now but also AIC and BIC are lower than the first model, we can conclude that these eleven tasks have a single dimension. Table 4.8 shows the result of CFA considering these eleven tasks. We can noticed that the factor loading are similar to that present in Table 4.7. We will discuss some issues regarding the factor loadings on next section.

Table 4.8: Confirmatory Factor Analysis of Instrument Two with Eleven Tasks

Task	Factor Loading (SE)	Task	Factor Loading (SE)
Traffic in the city	0.200 (0.037)	Meeting point	0.165 (0.042)
Storm proof network	0.183 (0.041)	Best translation	0.163 (0.041)
Footprints	0.158 (0.033)	Broken machines	0.151 (0.042)
Puddle jumping	0.174 (0.028)	True or false	0.168 (0.039)
Social network	0.193 (0.041)	Right rectangles	0.114 (0.041)
Height game	0.212 (0.041)		

4.5 Conclusion of Dimensionality

We can see that the results of multidimensionality do not support the allegation of more than one factor, neither statistical or theoretically. To illustrate, the factor loadings of CFA are negatives, as we have seen in Table 4.2. On the other hand, those tasks with factor loadings lower than 0.1 have shown poor loading, as we have observed in Table 4.4.

We have tried a new strategy redesigning different exploratory models, considering several combinations between tasks and abilities. Despite all efforts, none of the exploratory models seemed to fit. Therefore, we may conclude that this dataset did not fit the model of more than one factors.

Considering a theoretical approach, we also do not justify multiple factors. Indeed, we have conducted several exploratory studies [Araujo et al., 2018a; Araujo et al., 2018b; Araujo et al., 2018c] [1; 2; 3] over the years which have tried to name these factors presented in the exploratory phase of our study. However, we have not succeeded in finding the appropriated cognitive skills that are unique to each latent factor.

Moving on to the unidimensionality, we can support the allegation of a single factor, statistically and theoretically. As we have seen in Section 4.4, the group of tasks have demonstrated a good model fit and factor loadings. Although the factor loadings have been low to moderate, we will discuss why these values are expected, considering the context of Bebras Challenge and how the questions are designed in Chapter 6. For now, we have found that CT is the main latent trait under the tasks.

4.6 Summary of Chapter

In this chapter, we have presented the data screening of our study. We also have detailed the statistical procedures and results of factor analysis intended to demonstrate how many factors we can find in instruments. Finally, we have justified why we accepted the unidimensionality of both instruments.

Chapter 5

Evaluating Item Response Theory Properties in Computational Thinking Items

One of our objectives is to analyze how items have been designed to evaluate CT abilities. We address this issue adopting Item Response Theory as a method to appraise the items. Considering the overall advantage of IRT, we have focused on the property which provides information about the items. Therefore, we have centered the attention on two main criteria: the difficulty and discrimination parameters of items. These two elements are essential to assess how well an item can measure cognitive abilities in examinees [6; 7; 47; 68]. Therefore, we have gathered evidence to claim that (i) if we have predicted in advance the difficulty classification of items, the result may not be accurate, and (ii) IRT is a better choice to determine the difficulty level of the items.

This chapter presents the results of difficulty and discrimination level of items. Here, we are limited to show the graphics, numeric results, and classification of difficulty and discrimination based on IRT. In chapter 6, we will further detail on how the skills to solve the items and other features may influence discrimination and difficulty level.

5.1 Statistical Procedures of Item Response Theory

We carried out the IRT analysis using add-on Eirt for Microsoft Excel[1]. Eirt was developed using the Item Response Theory Library (libirt) which was written in C language under the terms of the GNU Public License. It is possible to estimate items parameter under one, two, or three logistic model (Rasch, 2PL, or 3PL) and abilities from examinees who responded to a test or a questionnaire. The parametric estimator is Bayes Modal Estimator and the ability estimator is Expected A Posteriori (EAP).

We have initiated the statistical procedures deciding the logistic model (Rasch, 2PL, or 3PL). The Rasch (or 1PL) model only considers the difficulty level, while the 2PL involves the discrimination and difficulty level. At least, the 3PL includes correct answers by guessing beyond both mentioned parameters. The following procedures will decide which model is acceptable. Table 5.1 shows the values of AIC and BIC when we have contrasted Rasch and 2PL models, as well as the 2PL and 3PL models for Instrument One, while table 5.2 shows the values for Instrument Two. We have chosen the model which presented lower values of AIC and BIC based on the protocol described on [Baker and Kim, 2017] [7]. In addition, we have conducted an analysis of variance (ANOVA) intended to know if the differences between the models are statistically significant.

Table 5.1: Fit of models - Instrument One

Model	**AIC**	**BIC**
Rasch	1724.207	1763.171
2PL	1717.394	1739.327

Model	**AIC**	**BIC**
2PL	1717.394	1789.327
3PL	1721.677	1829.577

Table 5.2: Fit of models - Instrument Two

Model	**AIC**	**BIC**
Rasch	2717.754	2768.299
2PL	2715.977	2710.749

Model	**AIC**	**BIC**
2PL	2715.977	2810.749
3PL	2724.253	2866.411

In both instruments, the 2PL model was the best fit model. Looking at Tables 5.1 and 5.2, we have observed that AIC and BIC present lower values at 2PL model. In addition, the result of ANOVA shows that there is a statistically significant difference between Rasch and

[1] Available at http://psychometricon.net/libirt/

2PL model (p-value = 0.002 for instrument One and p-value = 0.008 for instrument Two) but not between 2PL and 3PL model (p-value = 0.073 for instrument One and p-value = 0.115 for instrument Two), considering the significance level of 0.05. Therefore, we have conducted the IRT analysis using the 2PL model for both instruments.

5.2 Examining Items of Instrument One

We have started to analyze the IRT parameters by inspecting ICC in Figure 5.1. We have noticed that item 3, "Email" tasks, is not only the easier item (the most left-hand curve) but also one of item with the lowest discrimination value (ICC has a very small slope). We have observed that the probability of correct answer is up to 50% even for those who have low ability levels. In contrast, item 10, "Turn the cards", is the most difficult item and has also the lowest discrimination value due to the almost straight curve. We have seen the same curve behavior in item 8, "Fireworks" task, the ICC is nearly linear. Therefore, item 3, 8, and 10 are the items with the lowest discrimination values.

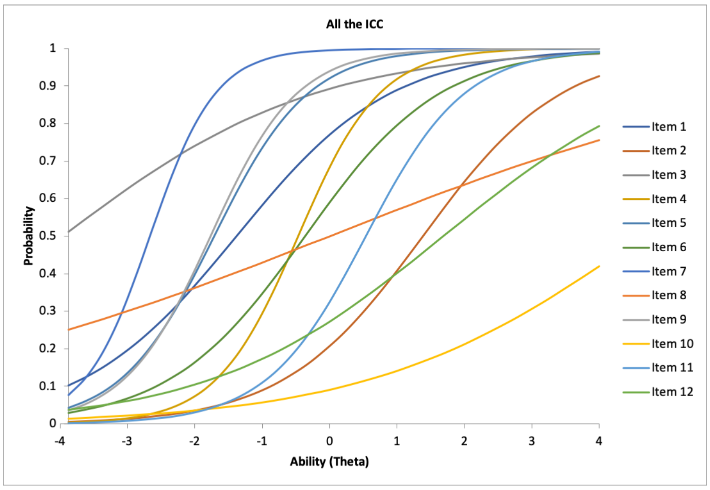


Figure 5.1: Item Curve Characteristics of Instrument One

Observing the discrimination a and difficulty b item parameters in Table 5.3, we can confirm the above-mentioned visual preview analysis and examine the other questions. Looking at difficulty parameter more carefully (column b), we have noticed two problematic items because they have critical values, i.e., the threshold value b is lower than -2.95 or higher than 2.95. Item 3, "Email tasks", is very easy ($b =$ -3.964). In contrast, item 10, "Turn the cards" is very hard ($b = 4.656$). Indeed, the ICC of those items are located more right and more left in Figure 5.1, respectively. Although Bebras organizers have classified "Email" task correctly and underestimate "Turn the cards" task, they are not good questions for assessing. "Turn the cards" task deals with logic reasoning implication, and maybe the implication was misunderstood. "Email" task deals with an obvious internet fraud to get money from naive people.

Table 5.3: Parameters of Instrument One

Item	Tasks	a	SE	b	SE
Item 1	Drawing stars	0.873	0.208	-1.38	0.33
Item 2	Bowl Factory	0.967	0.214	1.384	0.295
Item 3	Email	0.533	0.241	-3.964	1.683
Item 4	Beaver the alchemist	1.644	0.268	-0.466	0.122
Item 5	Tutorial	1.435	0.31	-1.712	0.277
Item 6	Popularity	0.995	0.2	-0.361	0.178
Item 7	Word chain	2.064	0.603	-2.664	0.446
Item 8	Fireworks	0.282	0.154	0.015	0.549
Item 9	You won't find it	1.558	0.337	-1.767	0.271
Item 10	Turn the cards	0.494	0.253	4.656	2.239
Item 11	Decorating chocolate	1.359	0.234	0.539	0.145
Item 12	Busy beaver	0.581	0.181	1.689	0.544

Contrasting the Bebras organizers' prediction of difficulty with IRT parameters, we have noticed only 58% of precise in difficulty prediction, as shown in Table 5.4. As we can see, five tasks are classified as easy (two of them very easy), four are medium, and three are hard (one of them is very hard), considering the interpretation of difficulty level by [Hambleton et al, 1991; Baker and Kim, 2017; Giacomoni et al, 2015] [42; 7; 36]. Despite the fact that Bebras organizers predicted almost of all tasks as easy and medium, they were wrong in precise which item would be. Thus, we can conclude that instrument One was easy to medium in difficulty level based on IRT.

Moving on discrimination analysis, at first glance, it seems that the instrument is balanced of all levels of discrimination. Table 5.5 shows the classification based on [Baker and Kim, 2017] [7]. We have seen that one item has very low discrimination value, while another one has very high. In the sequence, three items present lower discrimination as well as the other three show moderate discrimination. Finally, four items offer high discrimination values. Even though item difficulty and item discrimination are independent of each other, we can

Table 5.4: Difficulty of Item by IRT and Bebras Organizers - Instrument One

Item	Tasks	IRT	Bebras	Item	Tasks	IRT	Bebras
1	Drawing stars	Easy	Easy	7	Word chain	Very Easy	Medium
2	Bowl Factory	Hard	Medium	8	Fireworks	Medium	Hard
3	Email	Very Easy	Easy	9	You won't find it	Easy	Easy
4	Beaver alchemist	Medium	Medium	10	Turn the cards	Very Hard	Easy
5	Tutorial	Easy	Easy	11	Decorating chocol.	Medium	Easy
6	Popularity	Medium	Medium	12	Busy beaver	Hard	Hard

examine those parameters together in order to identify what kind of examinees the item can distinguish. Considering that the difficulty parameter poses the item on the ability scale (Theta) and the instrument has easy to moderate difficulty level, we can suggest that the instrument has moderate discrimination item for examinees with medium abilities (Theta).

Table 5.5: Discrimination of Item by IRT - Instrument One

Item	Tasks	Discrimination	Item	Tasks	Discrimination
1	Drawing stars	Moderate	7	Word chain	Very High
2	Bowl Factory	Moderate	8	Fireworks	Very Low
3	Email	Low	9	You won't find it	High
4	Beaver the alchemist	High	10	Turn the cards	Low
5	Tutorial	High	11	Decorating chocolate	High
6	Popularity	Moderate	12	Busy beaver	Low

Besides the discrimination level, we can examine the information function on Figure 5.2 in order to know how well the ability (Theta) is estimated [7]. Item 7 has the maximum amount of information on the ability level of approximately -2.5. This result means that the "Word chain" task can distinguish individuals with very low abilities. On the other hand, item 4 has an apex information on ability level nearly 0 (zero) and the ability ranges between -1.5 to +1.5. Items 9 and 5 have approximately the same amount of information for individuals with low abilities, while item 11 has information for medium to high abilities range $-1 < \theta < +2$. We have highlighted that from all questions, item 10 and item 3 have a lower amount of information in high and low ability level, respectively.

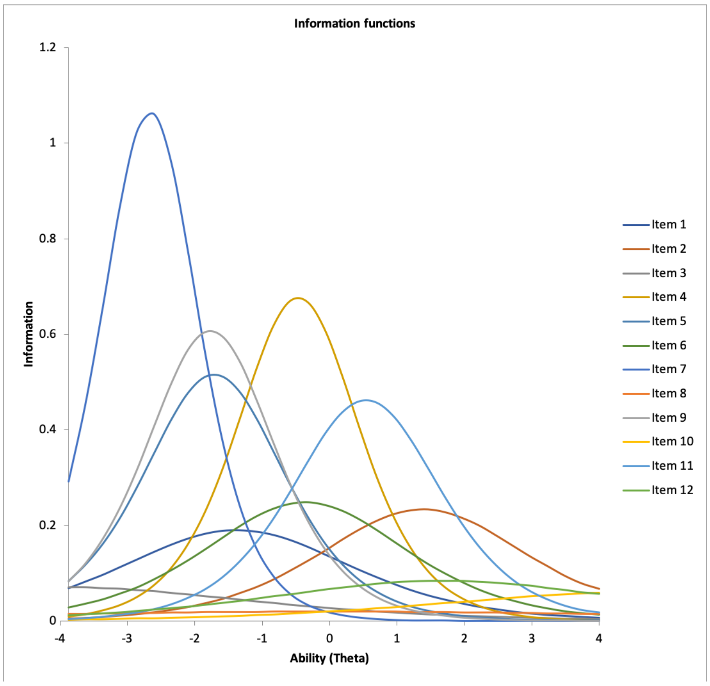


Figure 5.2: Information function of Instrument One

5.3 Examining Items of Instrument Two

Figure 5.3 provides an overview of the ICC to initiate the analysis of Instrument Two. Item 1, on the most right, "Ceremony" task, is the hardest question, while item 2, "Log-art" and item 3, "Beavers on the run" are the easier question. Items 1 and 2 have an almost straight curve indicating low discrimination values. Particularly, item 3 has the probability of almost 30% to be answered correctly by low ability individuals. Table 5.6 confirms the previews assumption.

Examining the difficulty values of items in Table 5.6, we have noticed that item 1, "Ceremony" task has a critical value (b =3.921), upper than normally expected (between -3 and +3). This result means that the question should be excluded from the instrument. Although item 2 has not critical values, the shape of the ICC curve in Figure 5.3 is nearly linear and should be reviewed too.

Observing the difficulty parameter, we can affirm that instrument Two has a medium difficulty level. We have seen in Table 5.7 that six items are easy tasks (three of them are very easy), seven are medium, and two are hard (none is very hard). Regardless of Bebras organizers' had classified five in each difficulty level, their categorization was not accurate.

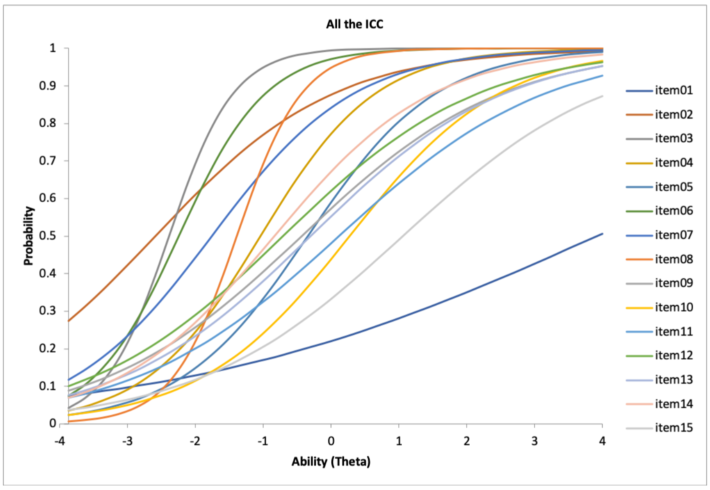


Figure 5.3: Item Curve Characteristics of Instrument Two

The precise in difficulty level prediction was merely 53% when contrast with the IRT.

Looking at the discrimination values, we can conclude that instrument Two has questions with moderate discrimination values. As Table 5.8 shows, only one item has very low discrimination value and another one has high discrimination value, while eleven items have moderate (73%), and two items have very high discrimination values. Considering that the item which presents very low discrimination is the same that should be excluded because it has critical values of difficulty level, the entire instrument offers items with moderate to high discrimination values. Since the instrument was classified as medium in difficulty level, we can suggest that the instrument can distinguish individuals with medium levels of ability (Theta).

When we have looked at the information functions of Figure 5.4, we can scrutinize which items have more amount of information in which ability level. The overall view is that items 3, 8, 6, and 4 have an amount of information under low ability. Meanwhile, items 5, 10, 12, 13 have an amount of information under medium ability scale. We have emphasized that item 1, "Ceremony" task has the lowest amount of information, close to zero.

5.4 Summary of Chapter

In this chapter, based on IRT parameters, we can conclude that instrument One has items with easy to medium difficulty level, whereas instrument Two has items with medium difficulty

Table 5.6: Parameters of Instrument Two

Item	Task	a	SE	b	SE
i01	Ceremony	0.321	0.161	3.921	1.918
i02	Log-art	0.759	0.214	-2.593	0.666
i03	Beavers on the run	2.103	0.523	-2.389	0.333
i04	Traffic in the city	1.172	0.208	-1.042	0.191
i05	Storm proof network	1.052	0.182	-0.345	0.153
i06	Space maze	1.556	0.358	-2.244	0.353
i07	Footprints	0.95	0.209	-1.762	0.352
i08	Puddle jumping	2.081	0.354	-1.39	0.152
i09	Social network	0.676	0.156	-0.437	0.234
i10	Height game	0.899	0.168	0.275	0.168
i11	Meeting point	0.653	0.153	0.118	0.22
i12	Best translation	0.692	0.159	-0.707	0.257
i13	Broken machines	0.697	0.156	-0.304	0.218
i14	True or false	0.852	0.173	-0.839	0.226
i15	Right rectangles	0.655	0.157	1.056	0.303

Table 5.7: Difficulty of Item by IRT and Bebras Organizers - Instrument Two

Item	Tasks	IRT	Bebras	Item	Tasks	IRT	Bebras
1	Ceremony	Hard	Easy	9	Social network	Medium	Medium
2	Log-art	Very easy	Easy	10	Height game	Medium	Hard
3	Beavers on the run	Very easy	Easy	11	Meeting point	Medium	Hard
4	Traffic in the city	Easy	Easy	12	Best translation	Medium	Hard
5	Storm proof network	Medium	Medium	13	Broken machines	Medium	Medium
6	Space maze	Very easy	Easy	14	True or false	Medium	Hard
7	Footprints	Easy	Medium	15	Right rectangles	Hard	Hard
8	Puddle jumping	Easy	Medium				

level. Bebras' organizers have demonstrated few accurate in predicting the difficulty levels when we have contrasted the results with the difficulty classification of IRT.

Regarding the discrimination and information functions, both instruments have items with low, moderate, and high discrimination values. However, instrument Two presents more significant number of items with moderate discrimination when contrast with instrument One. In chapter 6, we will discuss what can influence the difficulty and discrimination levels.

Table 5.8: Discrimination of item by IRT - Instrument Two

Item	Tasks	Discrimination	Item	Tasks	Discrimination
1	Ceremony	Very low	9	Social network	Moderate
2	Log-art	Moderate	10	Height game	Moderate
3	Beavers on the run	Very high	11	Meeting point	Moderate
4	Traffic in the city	Moderate	12	Best translation	Moderate
5	Storm proof network	Moderate	13	Broken machines	Moderate
6	Space maze	High	14	True or false	Moderate
7	Footprints	Moderate	15	Right rectangles	Moderate
8	Puddle jumping	Very high			

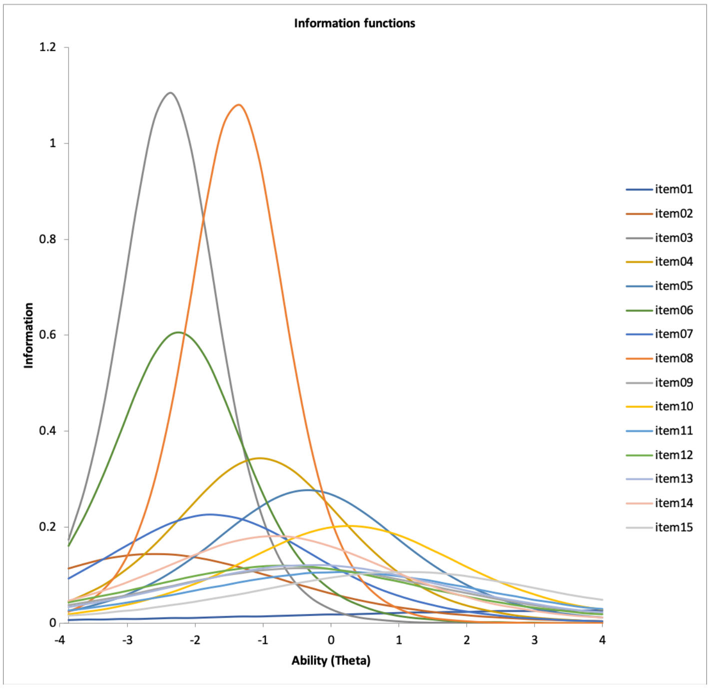


Figure 5.4: Information function of Instrument Two

Chapter 6

Discussion

In this thesis, we have proposed a new theoretical framework of CT skill based on empirical data. In this chapter, we have detailed the relations among how we solve the items and the results of factor analysis and IRT. Firstly, we have described the implications of CT factors and skills which inspire us to propose the theoretical CT model. In this context, we have explained how the items can be solved, associating them with difficulty level and discrimination based on IRT results. Then, we have gathered the lessons learned which arose from the conducted study. Lastly, we have presented some threats to the validity of our conclusions.

6.1 Computational Thinking as a Construct

We have discussed the relationship among factors, skills, and CT model. Then, we have analyzed the difficulty level and discrimination of items considering the proposed solution and the associated skills.

6.1.1 Factors, Skills, and Computational Thinking Model

In this section, we have explained the chronological order of the conducted research and discussed the implications of the findings in each step. Particularly, the chronological order of the events is important because it sums up how we have conducted the entire investigation of the observed skills, and how and why we have proposed the theoretical model.

Firstly, we have tried to confirm the claim of five abilities in the CT questions as Bebras Challenge suggested, but we failed. Bebras' organizers have suggested that the CT skill present in the items are abstraction, algorithmic thinking, decomposition, evaluation, and generalization [22; 11; 43]. Regardless of the source of the data, if we have collected ourselves or have used Bebras official dataset [3], none of the cases we had no statistical support of these five distinct CT skills. Indeed, Palts et al. pointed out similar results using a different Bebras official data [64]. This finding was a piece of evidence to look at CT skills more carefully, more precisely in how many those abilities are.

Secondly, we have followed a process to derive the number of skills in the CT question, like an exploratory study, which we have called multidimensional analysis. In one instrument, we found two main factors (Section 4.3.1), whereas in the other three main factors (Section 4.3.2). Pursuing the adopted process, we have attempted to name those factors, as follow. In instrument One, we have called the first factor *evaluation ability*, which would include abstraction, generalization, and decomposition, while the second factor was *algorithmic thinking and logical reasoning*. The same number of factors and skills were found in Bebras official data from Lithuania, considering the same questions, as we have carried on Araujo et al., 2019 [3]. In instrument Two, we have called the first factor *abstraction*, the second factor *algorithmic thinking and decomposition*, and finally, the third factor *algorithmic thinking and evaluation* [1]. However, after we have concluded all data collection and analyzed the data accordingly, we have reconsidered that it is not a proper classification because the questions have multiple associated abilities as well as the statistical results on commonalities of factor analysis pointed out limitations in this phase. The difficulty in interpreting factors can be justified due to low values of commonalities on PCA. The commonalities show how much of a question's variability can be explained by the set of factors extracted. Thereby, higher commonalities is better because the variability can better explain the set of factors extracted. Whether commonalities are low (values between 0.0 and 0.4), then a variable may attempt to load on any factor. In Tables 4.1 and 4.3, we have seen that the most commonalities are lower than 0.4, with few exceptions. These values means it is possible that some task is loaded onto another factor. For example, "Drawing stars" task has the lower commonalities. Besides, the factor loading is 0.47 on factor 1 and 0.29 on factor 2 (see Table 4.1). As we have used 0.30 as a sufficient factor loading [41], then, we have considered that "Drawing stars" task is loading on factor 1, but it was very closed to factor 2.

Thirdly, we have tried to confirm the number of factors found in the multidimensional study but we failed again. This result can be explained by reasoning that tasks are associated with more than one competence, consequently, more than one ability [22]. In fact, CFA is a statistical process that demonstrates how a set of observed variables are related to the underlying latent factor(s), considering that each variable was loaded into only one factor [41]. If the last condition is violated, the statistical results cannot support the allegation [15].

Despite the fact that we also have tried a different methodological procedure called Network Analysis in order to prove the multidimensionality, it was unsuccessful again. In the particular study we carried on [Araujo et al. 2018a] [1], we have used network analysis [47] in order to identify the relation among the questions of Instrument Two that shared the same abilities. We have observed three main factors, similar to the results of multidimensionality presented in Section 4.3.2. However, the associated question in each factor/construct are distinct, i.e., the tasks related to the factors in PCA are different from those related to the network analysis for the same set of tasks. This finding was another evidence that more than one factor did not fit on the analysed CT question.

Finally, we have moved on to a different perspective of how many and which skill are involved proposing a CT theoretical model. After the FA and IRT analysis, we have noticed that the existing CT models were not suitable. So, the theoretical model was slightly adjusted to achieve the desired fit. In other words, we only have considered the skills which we observed to solve the items based on the proposed solution by Bebras' organizers [11; 43].

The abilities and competencies proposed in our theoretical model are in line with the literature, but the organization and terms have revealed a new perspective. This new vision was motivated by the fact that the conducted empirical studies were not in agreement with the previous theories adopted in the literature. We have rethought not only the skills but also the action and high abilities involved in the resolution process. We were inspired in Barr and Stephenson study, which have claimed for nine CT capabilities: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation [8] [page 52]. We have highlighted that the authors called *capabilities* instead of *skills* or *abilities*. More than that, they have presented the capability and given some examples of how to apply it in K-12 disciplines. For instance, data analysis capability was explained as "Identify patterns for different sentence types" in Language Art, as "Identify trends in data from statistics" in Social Studies, and "Analyze data from an experiment" in Science [8]. In other words, just say *data analysis capability* is too vague because it can mean a lot of different actions. This study have suggested that the *capability* is in a high abstract level, which needs examples to show how to make it more concrete.

All things considered, our model is inspired by the idea of a distinct abstract level of cognition, considering that competencies and skills have different granularity. Instead of using the name of traditional skills or computational words, we have seen an opportunity to be more clear and transparent, adopting the specific competence and skill which the items assess. We have believed that if we endorse common words to name the skill, it can be more intuitive, easy to understand, and accessible to be applied by no-CS professionals. This approach is advantageous because no-CS teachers may face difficulty in adopting CT practices due to not have familiarity with CT vocabulary or not understand the essence of CT [60; 16; 89]. For instance, a recent survey have pointed out that teachers hold several views of the concept of CT [35]. Therefore, we have claimed that the CT model may help to disseminate CT in future works.

Lastly, we have emphasized that the correct approach is to propose a theoretical model first, and then design the items [47; 25]. However, our study has adopted an opposite approach, i.e., it has produced a conceptual model after the designed items, as a consequence of the main objective, which is to analyze the items. More than that, the preceding theoretical models have not supported the empirical data from our study. As a result, we have proposed a new framework of CT skills and competencies.

6.1.2 Difficulty of Items

Predicting the difficulty level of questions is complicated work. It depends on several aspects, such as the approach of stakeholders who have proposed the instrument, the tacit knowledge of test's designers, the subjects who will be submitted to the instrument, and the characteristics of the questions [85; 26]. Despite all effort, none of the cases is entirely reliably. The best way to find the difficulty of questions is to examine the results of the test than trying to anticipate [7; 25; 47].

The Classical Test Theory (CTT) affirms that the facility/difficulty level is determined by the total amount of correct/incorrect answers in an instrument. For example, the rate of the facility is calculated by the division of the correct response to a question and the total number of subjects [32]. This result represents the percentage of correct answer ranging from 0 to 100%. The higher the percentage, the easier the question. For instance, above 75%, the question is classified as easy and indicates that many examinees get the correct answer. In contrast, lower to 25%, the question is categorized as hard and shows that few subjects endorse the correct answer [32]. In addition, the raw test score is the sum of the scores received on the items which reveal the overall result of each student based on his/her location at a standard scale for the particular test [25].

Under Item Response Theory, the difficulty level is measured at the ability scale (Θ) when the probability of correct response is 50% [6]. Remembering that, the graphic of ICC shows the relation between the probability of endorsing the correct answer and the ability scale (Θ) (See chapter 2, section 2.5 for more details). If an item is categorized as easy, this point appears at a low ability level. In contrast, if an item is classified as hard, this value occurs at a high ability level. As we have adopted the IRT, we have no interest in the score of the instruments to discuss the difficulty of questions. Therefore, all the discussion about the difficulty level is based on the difficulty parameter under IRT instead of scores.

Our result suggests that Bebras' organizers have achieved merely 50% of accuracy in predicting the difficulty level (see Chapter 5). This particular finding reinforces the idea that anticipating the difficulty of questions is a complex activity. In the context of Bebras Challenge, there is not a problem because it is a contest whose solely goal is to promote CT. However, our data suggest that if Bebras' tasks will be used as a measuring instrument, more investigation should be taken.

In the direction of examining the tasks, we have started to look into the tasks with extreme values, for example, very easy, very hard as well as critical values of difficulty parameter. Then, we will see the other questions and discuss discrimination in the next section.

"Email" task is one of two tasks that has no figure but we have believed that is not the principal issue. Although the question is about how the internet can be used for criminal people to receive money from innocent ones, it has a text to interpret and judge based on your familiarly to real-world values. No doubt that internet safety is an important content nowadays and the solver explores evaluate skill, but it seems that it is not useful to assess CT because it is classified not only as a very easy task with critical value but also has low discrimination value (see Tables 5.4 and 5.5).

As the opposite, "Turn the card" is classified as the hardest question in Instrument One, even it is necessary only logical reason to solve it. At first glance, the solver is misled into thinking that it is only required to move the vowel and the even card. The statement can be seen as a distractor. However, if the solver has known the true table of logical implication, it increases the probability to endorse the correct answer: the vowel and old cards. Besides, the solver should think that he/she does not need to flip the 2 card because it is not forbidden to have a consonant on the other side. However, if there is a vowel on the other side of 7 card, the statement is false. All things considered, tasks like this should be excluded because it has critical value on difficulty parameter.

The easier questions are "Email", "Word Chain", "You won't find it", and "Tutorial" in Instrument One. We have noticed that "Email" and "Tutorial" tasks have no figures to illustrate the problem situation. Notably, they are the only question without illustration in both instruments. Even without a draw to exemplify the question, these tasks are categorized as easy because both explore the evaluation ability and "Tutorial" also explores to obey rules skill. On the other hand, in "Word Chain" task, the figures are essential to solve the question. In this case, the solver just needs to pay attention to the pattern in the illustration, identify similarities, and choose a representation of the solution. It was a visual perception question. Finally, "You won't find it" task has demonstrated how to follow a pre-defined algorithm in reverse order using a flowchart. Probably the visual representation of the diagram turns it accessible to apply the algorithm. The solver only needs to know the alphabetic order as a preview knowledge.

When we have looked at the easier tasks on Instrument Two, we have found "Log-art", "Beavers on the run", "Space maze" which were classified as very easy questions. "Log-art" and "Word chain" tasks have explored visual perception. The solver should find the pattern at the begging of the question and choose the alternative that is part of the pattern. On the other hand, "Beavers on the run" task encompasses to learn an algorithm and apply in a similar but more complicated situation. No external knowledge is needed to solve these questions. Last but not least, in the "Space maze" task, the solver should associate the instructions on the alternatives with the directions which a robot must follow to reach the goal. Although this last question has required that the solver knew north, west, east, and south direction, it was an easy task.

Meanwhile, instrument Two has only two hard tasks, "Ceremony" and "Right rectangles". Specifically, "Ceremony" is classified as very hard task with critical value (see Table 5.6). We have believed that this result is explained to the fact that it is an order question but free-response item. When we were correcting the tests, we have noticed that many students were confused in writing the numbers in order and mixed up few alternatives. Maybe an order question with eight steps to put in sequence should be rethinking to a way that the solver writes down the correct answer without distractions. Then, "Right rectangles" is categorized as hard task because it encompasses single and complex instructions, as well as the spacial and math notion, like, "turn 90°clockwise" n-times. This is one of the best questions to assess students with high abilities (Theta) in CT.

Finally, when we have placed the results from factor analysis and IRT together, we noticed that they are related. In chapter 4, section 4.4, we have observed that five tasks in instrument One and four tasks in instrument Two have shown factor loading lower than 0.1; consequently, they should be dropped off. Likewise, under IRT, these items have presented critical values of difficulty level or discrimination; consequently, they are not useful for assessment and should be removed. For instance, in Instrument One, "Email" and "Word chain" tasks are classified as very easy, "Fireworks" and "Busy beaver" tasks have low discrimination, "Turn the cards" is categorized as very hard (with critical value) as well as has low discrimination. In instrument Two, "Log-art", "Beavers on the run", and "Space maze" tasks are classified as very easy, while "Ceremony" task as very hard (with critical value) as well as has very low discrimination. Therefore, the same items have demonstrated critical issues considering results from factor analysis and IRT.

In summary, different from classical test theory, under IRT, the difficulty parameter is a location index. Then, an instrument needs easy, medium, and hard question to cover all the ability scale. Thus, the instrument can assess examinees with different ability levels. In our study, we have noticed that both instruments have easy and medium tasks but few hard questions (see Tables 5.4 and 5.7). So, both instruments, as they are now, may assess individuals with low and medium level of CT skills. However, they may not measure students with high abilities.

6.1.3 Discrimination of Items

Remembering that discrimination describes how well a question can distinguish examinees proficiency, we will analyze the tasks as follow. Both instruments have shown question in all scale of discrimination. We also have observed very low but not negative discrimination values which is expected.

In instrument One, items with high discrimination values and classified as medium difficulty are "Beaver the alchemist" and "Decorating chocolate" tasks. We can solve the first one by counting the objects and tracking the rules of transformation. The graph helps us to visualize the text rules, and we need to sum the total amount of objects. The second one is a classic algorithmic thinking task which describes the commands. In addition, the entire code is shown in the four multiple-choice alternatives. However, the solvers have needed to select the incorrect option. Besides, the solver must know geometric angles to settle the task.

Items with moderate discrimination values can be observed in "Bowl factory" and "Popularity" tasks. The first one explains how to order bowls with different sizes using the bubble sort algorithm. It asks how many iterations are needed to put in sequence a set of different size of bowls. Particularly, this question has good discrimination value on high ability level. The last one is a question with a graph relation. We need to count the most direct and indirect node (one step). This rule is described in the text, and the relations are shown in the picture.

In instrument Two, we already have discussed the tasks classified as very high and has high discrimination value in Section 6.1.2 because they are categorized as very easy or easy question too. Now, we have focused attention on the task with moderate discrimination value classified as medium difficulty. "Storm proof network" task shows a network topology which the towers can be the nodes and the way they are connected can be seen as a graph. The solver needs to observe which node will interrupt the connection.

The tasks with low discrimination value have some issues that should be highlighted. "Email" and "Turn the card" tasks have critical values on difficulty parameter, i.e., they are trouble tasks. In "Busy beaver" task, the figure seems not to help to interpret the text rules, in contrast with those tasks with proper discrimination. More than that, the rules are confusing, and the table appears not help to aggregate the statements. Besides, it is an open question, i.e., there are no choices. Finally, although "Fireworks" task is a visual perception question, it maybe has a distractor. The solver is misled into thinking that it is only three combinations (the most frequent answer), whereas it is four possible combinations. Maybe the students do not know how to convince yourself that there are no more possibilities of arrangements using a systematical approach instead of counting.

6.2 Lessons Learned

6.2.1 Computational Thinking Items

Items are the central unit of every instrument. Our attention in this investigation was in how these items express the CT skills in their solution process. We have gathered some patterns found on the analysis of the instruments as follows.

1. Items with figures demonstrate satisfactory discrimination. We have noticed that most of the items present illustrations. The figures can *(i)* explain the problem, or *(ii)* demonstrate the problem statements, or *(iii)* help to understand the steps. However, just a general illustration may not assist in increasing discrimination. We have noticed that the figure should assist solvers in understanding the problem and creating a mental representation of it. Thereupon, illustrations and examples which depict the problem or solution may help increase discrimination.

2. Items with graph and text rule present excellent discrimination. Graphs are a type of data structure applied to demonstrate the relationship among elements. It also can be used to make it easier to visualize connection, algorithms, and rules. We have observed when graphs are related to text rules, the items offer redundant information, i.e., the figure represents the text rules. Thus, we have believed that not only motivated and careful solvers can settle the problem but also those who not pay attention to the text description. Another interpretation is that if the solver can see the illustration of a text rule, they can better understand how to apply the rule by mental representation. Therefore, it is essential to associate text rule with graphs whenever possible.

We also have highlighted that it is possible a graph was not related to executing an algorithm. For example, the "Storm proof network" task has a medium difficulty level with moderate discrimination and is not associated with a descriptive algorithm. It shows a representation of a topology network which can be interpreted as a graph. However, there is no algorithm to follow, just one rule to obey and check (evaluate) careful in each alternative.

3. Items with logical statements should be carefully thought in order to have good discrimination. We have observed two types of item using logical statements. One have showed medium difficulty with moderate discrimination, and the other has revealed low discrimination value and very hard difficulty level. The main difference was that the first, "True and False" task, used only logical statements while the second, "Turn the card" used one rule of the true table of logical implication. We have believed that using a specific rule of the true table was not a good strategy to design an item which does not require any previous knowledge. This claim arises from a question ("Turn the card") which can not distinguish students with low abilities and only works for who have very high-level skills. For this reason, the designers of the items should be careful thought considering what type of logical statements they have requested.

4. Items which explore executing an algorithm skill *with commands present good discrimination in all ability scale.* When we have gathered the tasks with explicit commands which request to run an algorithm, we have noticed that they cover all ability scale (Theta), and presented moderate and high discrimination values. This behavior occurs with "Right rectangles", "Broken Machine", "Decorating chocolate", "Footprints", and "Space Maze" tasks. Therefore, items which encompass algorithm with commands are a great option to design questions with good discrimination and cover the ability scale.

5. Recognizing sub-problems (decomposition) can be associated with two kinds of solutions: recursion or establishing intermediate results. In the first case, the previous answer composes the next answer systematically until the final response (recursion). For example, in "Footprint" task, the final solution depends on the answer of smaller instances of the same problem statement, such as drawing the footprint tree. In the second case, the sub-problems can be different or have distinct answer but they are essential to reach the final solution. To illustrate, in "Height game" task, the solver should recognize that there are two sub-problems. At the beginning of the resolution, the solver needs to order the characters by height. Lastly, he/she has applied the restrictions (rules) to order the characters as the instructions say. So, the designers of the items should know what type of decomposition they have requested to solve the problem.

6.2.2 Designing Computational Thinking Assessment Instruments

We have deemed important to register some lessons learned during the research on trying to find a reliable instrument to measure CT. The directions described here came from all the investigation on how to assess CT skills through the adopted instruments and the psychometric literature [7; 25; 47; 68; 4].

1. It is crucial to follow a consolidated methodology. A robust method provides a security guideline which teaches us how to produce a trustworthy instrument. Thus, it will detail all the steps to follow in order to design a reliable tool. In addition, the methodology should be satisfactory to the research objective as well as the type of desirable instrument. We claim that IRT is a robust and practical approach to design CT assess instruments. As we have detailed on [4], it is possible to pursue steps in order to produce a reliable test. Besides, IRT parameters can provide useful information regarding difficulty level, discrimination, and guessing by chance.

2. The concept of CT and the associated skills explored in the assessment instrument should be as precise as possible. It is essential to clarify the definition of CT as well as the skills explored in the instrument. Those specifications will guide the items designated to measure the skills. The abilities also should be detailed, considering the action to solve the questions. If these concepts are not established previously, before the design of the items, the whole instrument can be wrong.

3. A group of CT specialists should evaluate and judge not only the content but also the associated skills in the instrument. The analysis of specialists is a common practice in psychometric studies in order to ensure semantic and content analysis [68; 67]. Concerning the content, the specialists (or trained raters) can decide whether the items are intelligible, i.e., the problem statement can be read and understand by any individual, and there are not elements which embarrass or distract the solver. Regardless of the CT skills, the trained rater needs to observe if the items explore a problem that can be settled by CT skill. Then, she/her should claim the associated skills explored during the resolution process.

In Computer Science and Education, we have faced some limitations that should be addressed by future work. First, evaluating items is not a trivial activity. Generally, it is not a competence that computer scientists have learned, even if they have followed the educational field. Different from the psychometric area, we do not have a network of researchers able to act as trained raters in cognitive contents. Second, there are few researchers in CT, mainly in Brazil. Fortunately, in the last years, we have counted on a national workshop called WAlgProg (Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação) to gather studies in CT since 2015. However, most of the published studied claim that teaching programming is a synonym of stimulating (and assessing) CT skills. We have believed that programming is an approach to promote CT, but should be carefully analyzed as cognitive ability in order to better understand how it is related to Computer Science itself, programming techniques, and other fields, like mathematics. Third, we do not have tools or systems to assist us in executing the activity of evaluating items for assessment. Therefore, these issues were a limitation of our research.

We have advocated for a network of specialists in Computer Science Education in order to improve the reliability of assessment instruments. The specialists in several CS Education areas should be guided in how to act as a trained rater in their expertise. This approach can increase the quality of how we assess the students in our research and the confidence of the claims.

4. Predicting the difficulty level of items may not be accurate if we considered the theory or tacit knowledge of organizers exclusively. The difficulty of items can have different perceptions, depending on the strategy of classification, before or after the application of the items. Whether the tacit knowledge or theory were used as criteria for the rating (before the application), the difficulty level can be under or overestimated. In our study, we obtained 50% of accuracy approximately, considering the predicted by Bebras' organizers (before the application) and the results of IRT (after the application). Therefore, it is more accurately appraise the difficulty level after the students solved the items.

In addition, item difficulty may be intentional or accidental. In the first case, the designer can premeditate complicate elements intended to the solver address them. In the last case is the opposite; the difficulty is unintentional and undesirable. Frequently, the unforeseen difficulty only can be observed after the application of the items. In our study, we have observed that in some tasks, mathematics content is needed, such as elementary arithmetic and basic geometry angles, but it does not seem the main difficulty when settling the problem. In fact, those items have presented moderate and high discrimination as well as easy, medium, and hard difficulty level, which is desirable. We have believed that the mathematical content is elementary considering the individuals of our study, neither it was complex or challenging. Probably, for assessing children, this point should be carefully rethinking because we may underestimate such mathematical difficulty.

Lastly, even following all protocol of methods, it may be imprecise to define the difficulty considering the theory or tacit knowledge exclusively. It is essential that experimental studies systematically manipulate items and eventually improve, change, or remove them from the instrument based on empirical evidence. Then the questions are tested and retested with target subjects. At last, the instrument will be calibrated with easy, medium, and hard items as desirable.

5. Measuring skills without a total score of the correct answers can bring benefits to CT. The usual way to assess abilities is by the overall score of the correct answers in an instrument, as Classical Test Theory advocates. This approach provides two main limitations. Firstly, if two individuals have the same number of the total score, they have the same ability level, even they endorse items with a different difficulty level. Secondly, the result depends on the sample of individuals who have been examined. Whether we have designed instruments without the above-mentioned limitations, we can advance in the CT field and better understand the cognitive abilities involved. In addition, it will be possible to comprehend the relation between CT and other areas, like Mathematics and Engineering or even cognitive skills, such as intelligence and creativity. To mitigate these limitations, IRT is an appropriate methodology.

6.3 Threats to validity

As happen in all experimental research, some issues threaten the validity of our study. Concerning the internal validity, as our work involved cognitive assessment, human factors threaten its validity. We have collected data from groups twice in a semester. Thus, the time can influence how individuals reacted to the test at the beginning and the end of the semester. Although we have carried out statistical procedures to ensure that the groups can from the same population, we did not select a random sample of students. Instead, we have invited all student enrolled in an introductory programming course to participate voluntarily.

As for construct validity, we have highlighted that we applied the concepts and definitions of CT and CT skills claimed by the literature. However, some items can be answered exploring distinct abilities from those which they were designed to. In other words, different students may vary the way they solve an item. In order to mitigate this threat, we have named the skills in our theoretical model as actions, more operational as possible.

Related to external validity, although we have involved students of three different undergraduate program from two universities, the results could not be generalized to every undergraduate student. However, we have attended the minimal number of sample to ensure the statistical value in factor analysis and IRT procedures. Despite the overall threats, the idea and research methodology used in this work can be applied in other instruments and other students in different level of education.

Finally, we have highlighted some threats related to the theoretical model. Firstly, the model was designed after the evaluation of items instead before. Considering a psychometric approach, the appropriate order is to propose a model and then create the items. However, our study has adopted the opposite approach, i.e., it has produced a conceptual model after the investigation as a consequence of the main objective, which is to analyze the items. Secondly, the model is related to the competencies and skills exploited in Bebras Challenge. In this case, CT abilities are affected by the Bebras organizers' approach. Lastly, we have not executed a psychometric validation of the model yet. We will perform psychometric validation in a future work.

6.4 Summary of Chapter

In this chapter, we have discussed how the solving process are related to the results of discrimination and difficulty level pointed out by IRT analysis. During this process, we have gathered some patterns of associating characteristics and abilities with the results of discrimination and difficulty level. For example, we have learned that items with graph and text rule present excellent discrimination. Besides, the algorithmic skill was an essential ability to be explored in items because it has presented good discrimination in all ability scale. At this moment, we also have noticed some advice on how to design a trustful assess instrument. In addition, we have claimed for a network of CS educators in order to assist the validation of tools. All things considered, we have underlined some threats to the validity of our study.

Chapter 7

Related Work

In this chapter, we have presented some studies in CT assessment, which are in line with our proposal in this research. We have divided the discussion into three perspectives. Firstly, we will present some CT model proposed by other studies and contrast to ours. Secondly, we will detail studies which are dedicated to analyze CT items similarly as we did. Finally, we will present some instruments and their limitations.

7.1 Computational Thinking Model

There are several CT models in the literature to asses CT skills, mainly based on programming practices. Brennan and Resnick have proposed a CT framework which is composed of three different dimensions: Computational concepts, Computational practices, and Computational perspectives [14]. Other approaches came from their idea, such as Zhong et at. carried on [92]. They have thought in a model involving three distinct areas of computing related to programming practices. Following a similar line, Computational Thinking Patterns are abstract programming skills explored in a game programming environment called AgentSheets guided by the Scalable Game Design group [9; 52]. Generation, Absorption, Collision, Transportation are examples of Computational Thinking Patterns in a game [9]. However, these proposals are used in a specific programming environment, Scratch[1] and AgentSheets[2], respectively, different from our model, which

[1]Scratch. Available in `https://scratch.mit.edu/`

[2]AgentShheets. Available in `https://www.agentsheets.com/`

was designed for no-programming practices.

Educators associations in computing have proposed theoretical models. A well-known CT model is suggested by Barr and Stephenson in association with CSTA and ISTE. The model have pointed out a set of CT abilities to K-12 students: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation [8]. Computing at School, a UK association, have suggested a model of five skills encompassing abstraction, decomposition, evaluation, generalization, and algorithmic thinking [18]. The New South Wales Department of Education in Australia[3] has proposed that CT includes abstraction, decomposition, data organize logically, and designing algorithms, patterns, and models. At first glance, these models are theoretical proposals intended to guide teachers to apply CT in practice. Moreover, these proposals have incorporated or inspired the curriculum of K-12 computing education around the world. Similar to the theoretical models proposed by educators associations, we have believed that our model has the potential to be used in future works as well as encourage teachers to apply the CT skills in their practices at classrooms.

There have been several literary style reviews which suggested CT skills models. Selby and Woollard have analyzed the most frequently occurring skills in CT literature from 2005 to 2015 [81]. They have found that CT encompasses a thought process of abstraction, algorithmic design, evaluation, decomposition, and generalization [81; 83]. Notably, this research is vital because it have inspired the Computer at School and Bebras Challenge to propose their CT skills models as well as it was the first model we adopted in our doctoral study before we have introduced our theoretical model. Following the same idea, Kalelioglu, Gulbahar, and Kukul also have presented a framework of CT skills based on systematic research review. Their study have pointed out the most commonly fifteen used skills in the definitions of CT [50]. They have summarized the top five skills related to CT, which are: abstraction, algorithmic thinking, problem-solving, pattern recognition, design-based thinking. Gouwn, Bradshaw, and Wentworth have examined several definitions of CT skills in literature and grouped the skills and practices into six distinct classes [38]. The

[3] `https://education.nsw.gov.au/our-priorities/innovate-for-the-future/education-for-a-changing-world/media/documents/future-frontiers-education-for-an-ai-Coding-and-Computational-Report_A.pdf`

classifications of CT skills and practices based on Gouwn, Bradshaw, and Wentworth are: *(i)* processes and transformations; *(ii)* models and abstractions; *(iii)* patterns and algorithms; *(iv)* tools and resources; *(v)* inference and logic; finally, *(vi)* evaluations and improvements.

Different from our study, these above-mentioned proposals were theoretical or were aroused from the literature review. Our CT model have resulted in a combination of empirical data, state of the art in CT, and elements of cognitive assessment. The skills and competencies present in our model have been explored in state of the art in some way, i.e., we do not create new abilities in CT. In fact, we have expressed the CT skills in a fresh perspective considering the Vocabulary and granularity of competencies and skills.

7.2 Analyzing Bebras' Items

The earlier analysis of Bebras' tasks is related to difficulty prediction. Van der Vegt has examined the difficulty of Bebras tasks by success rates [85]. In a recent study, he also has analyzed the difficulty of tasks in the Netherlands using elements of the cognitive load theory [86]. Yagunova et al. have discussed complexity and difficulty of tasks in Russian contest concerning the number of correctly solved tasks and their mental workload to be addressed, such as cognitive, informational, and emotional loads [91]. Pohl and Hein have analyzed some tasks and proposed recommendations to improve the quality of tasks presentation, like short sentences, appropriate analogies, precise definitions, and unambiguous wording [70]. These studies have examined the difficulty of tasks using a classical approach, i.e., looking at the total score of correct answers. In our research, we have avoided this methodology and have applied IRT analysis.

Other studies have applied IRT to analyze Bebras' items as a reliable approach intended to explore the potential of the tasks to assess CT. Lonati et al. not only have examined the tasks of the official section of Bebras Challenge 2015 in Italy under IRT but also have contrasted the results with a new section with some modifications in the tasks [58]. Bellettini et al. have compared the difficulty level of Bebras Challenge 2014 predicted by organizers in Italy with difficulty faced in IRT analysis after the contest [10]. Both studies have concluded that predicting the difficulty level of tasks is far from being a trivial activity. Dolgopolovas et al. have used Bebras tasks under IRT analysis trying to measure CT skill in novice software engineering students [27]. Hubwieser and Muhling have developed a specific method of applying distinct mathematical models of IRT in order to explore tasks in Bebras Challenge 2009 in Germany [46]. They also have identified differences between the performance of boys and girls.

Particularly in the Bebras Challenge, neither the difficulty or discrimination of tasks is a primary concern of tasks' designers [58; 85]. Because of that, the studies mentioned above have explored the psychometric parameters towards measuring CT abilities. Likewise, we have concerned about the difficulty and discrimination level of items, and we have carried on an IRT analysis on Bebras' tasks. However, different from the previous works, we have analyzed the common features of how the students can solve the items carefully. Then, we have gathered which characteristics can be seem like common constructs. In other words, we have saw which elements in a task may influence the difficulty and discrimination level. We have detailed these elements as lessons learned (see Section 6.2).

7.3 Instruments

Although our study is focused on items, we have deemed important to discuss some instruments used to assess CT skills, specially those works who explore Bebras as an test of measuring CT abilities. Dolgopolovas et al. have selected Bebras tasks to build a test for assessing CT skill among the first year software engineering students in a structured programming course at Vilnius University [27]. However, the results have shown no correlations between the Bebras' test and the grades in structured programming. Similarly, in the beginning of our doctoral study, we also have explored Bebras as instrument intended to correlate the Bebras' scores with grades of introductory programming course in novices CS students [5]. We found positive moderate correlations between Bebras' scores and programming grades in both Bebras' tests applied at the beginning and the end of the semester. In the same way, Matsuzawa et al. also have used the Bebras Challenge as a measurement instrument of CT skills in an introductory programming course designed for non-CS undergraduates [61]. They have correlated the result with practical and paper tests of programming. The result have indicated a weak positive correlation between the three tests. Lockwood and Mooney have developed two CT test based on the tasks developed for the Bebras Challenge [56]. They have affirmed that both test have equivalent in difficult level based on the validation conducted by teachers, undergraduate and post-graduate students. Despite that, they found no significant difference between the test applied before and after programming classes. Roman-Gonzalez, Moreno-Leon and Robles have contrasted some selected Bebras' tasks with a validated instrument to measure CT skill, named *Computational Thinking Test - CTt*, in order to address a convergent validity [77]. They have found a statistically significant, positive, and moderately correlation which reinforced the idea of CTt is partially convergent with Bebras tasks. Likewise, Boom et al. have used Bebras as an assessment of CT and an intelligence test to measure general problem-solving skills. The result have indicated a large and significant correlation between these two instruments [12].

Similar to these studies, we have investigated how Bebras can be taken as an instrument to assess CT abilities. As highlighted by Dagiene et al., the primary goal of the Bebras Challenge is not to test students' knowledge or CT abilities, but with more studies, it looks like that Bebras can be used for assessment [23]. However, our findings in this research have suggested that it is not very likely that CT measures can be derived from the Bebras as it is currently designed. Therefore, to use Bebras Challenge as a reliable instrument, further modifications of original design should be conducted.

Different from those tests based on Bebras Challenges, Gouws, Bradshaw, and Wentworth have created their on instrument to assess CT skills. [38]. Items of the Computer Olympiad Talent Search of South Africa were the source of the test. Thus, they have selected the items based on their definition of CT skills model. Recently, the Education Government of Australia have proposed the Computational and Algorithmic Thinking (CAT) competition[4]. The test is problem solving competition which the goals is promote multiple modes of thinking, especially CT. The first edition run on April 2019.

Aside from the previous instruments, we have highlighted two validated test to measure CT skills, but they assess different CT abilities. The first is the Computational Thinking Test (CTt) which addresses computational concepts, such as basic directions and sequences, loops, repeat, and conditional statements [76]. It is a multiple- choice instrument composed by 28 items, which are administered on-line intended for Spanish students between 12 and 14

[4]Computational and Algorithmic Thinking (CAT) competition. Available in: `https://www.amt.edu.au/department/cat-information`

years old. The second is the CT Scale [53] which encompasses five CT factors: algorithmic thinking, critical thinking, creativity, cooperativeness, and problem solving. It is a five-point likert scale composed by 29 items, for graduate and undergraduate students (adults). Both are independent of programming environments, but the first test assess skills related to programming and algorithmic problems while the second is associated with generic cognitive abilities.

Developing appropriate instruments to measure students' cognitive abilities as CT skills is not an easy endeavor; moreover, it is an incipient way to Computer Science field. According to psychometric, a test requires a rigorous process before it can be effectively used as a measuring instrument. Regarding of CTt and CTScale, to the best of our knowledge, the other instruments did not present validity or reliability. Therefore, the results can not be totally scientific trustful.

Chapter 8

Conclusion

In this work, we have investigated strategies and instruments in order to quantify CT skills reliably without mandatory programming practices. The challenge was to choose a suitable methodology to guide us in a well-founded analysis in order to understand the nature of the cognitive abilities involved in solving problems in CT. This study has encompassed a complete analysis of CT instruments considering statistical methods not only to identify CT factors but also evaluate psychometric properties of items.

The main contribution of this thesis is to evaluate strategies intended to quantify CT skills as a cognitive ability. We have proposed a theoretical CT model which encompasses skills and competencies. This model has detailed which skills we can observe and measure in problems without computer programming activities. The conducted analysis has facilitated to identify CT as a single construct but demanded other skills and, more abstractly, other competencies. Thus, we can address the relations of skills and competencies and express them in an abstract model.

The proposed model has represented the observed skills in our study. In summary, we have defined that CT is an approach to solve well-defined and knowledge-lean problems supported by four competencies and twelve skills. The first competence, *understanding the problem competence*, encompasses to systematize the problem indented to better understand what the question asks. The associated skills include *highlighting the essential data*, *recognizing sub-problems*, and *thinking like component parts*. The second competence, *analyzing the data competence*, involves to plain how to find the solution by examining the problem statement. The associated abilities comprise *finding patterns*, *identifying similarities and* *connections*, and *evaluation*. The third competence, *thinking about instructions competence*, includes to act based on the requirements or make a judgment about a situation after considering the available information. The associated skills encompass *executing an algorithm*, *obeying the requirements*, and *developing suitable solution*. Finally, *proposing a solution competence* involves to systematize the data during the resolution process and get directions in order to achieve the answer. The associated skills consist of *choosing a representation of a solution*, *using a previous solution*, and *recognizing a solution*.

We have perceived that if we want to measure CT skill singly, we should design questions which encompass the aimed ability. The first step in this direction is to settle on a well-founded theoretical model. This model demonstrates what will be measure. In other words, the model communicates, without ambiguity, which cognitive abilities should be exploited to settle the problem. Likewise, items should be designed to focus on mandatory skill (or single). We have noticed that it is a challenging effort because CT problems often involve more than one skill. For example, the ability to execute an algorithm can be associated with the ability to recognize sub-problems or/and evaluating steps in order to make a decision in which direction to follow. More than that, the ability to recognize sub-problems may not solve the problem by itself; it is needed another skill to answer the question unless the item only requires to identify the sub-problem.

We have contributed with a set of lessons learned in order to assist future research in design items to measure CT skills reliably. Above all, we have advocated following a consolidated methodology intended to create trustworthy instruments. In this direction, the concept of CT and the abilities should be as precise as possible. Moreover, experts in CT should evaluate not only the content but also the associated CT skills in the instrument.

Finally, we have carried on several studies through this research. In this document, we gathered the main findings. Over the years, we have published the partial results and other contributions. The publications in chronological order are the following:

- *A Systematic Mapping Study on Assessing Computational Thinking Abilities* in Frontiers in Education (FIE) 2016.

- *Um Mapeamento Sistemático sobre Avaliação do Pensamento Computacional no Brasil* in Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg) 2016(in Portuguese).

- *Exploring Computational Thinking Assessment in Introductory Programming Courses* - Frontiers in Education (FIE) 2017
- *Explorando Teoria de Resposta ao Item na Avaliação de Pensamento Computacional: Estudo em Questões da Competição Bebras* in Brazilian Symposium on Computers in Education (SBIE) 2018 (in Portuguese).
- *Análise de Rede na Indentificação de Habilidades Relacionadas ao Pensamento Computacional* in Brazilian Symposium on Computers in Education (SBIE) 2018 (in Portuguese).
- *Como Identificar Habilidades do Pensamento Computacional? Um Estudo Empregando Análise Fatorial* in Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg) 2018 (in Portuguese).
- *How Many Abilities Can We Measure in Computational Thinking? A Study on Bebras Challenge* in Special Interest Group on Computer Science Education (SIGCSE) 2019.
- Chapter of book: *Teoria de Resposta ao Item* in Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa. Porto Alegre: SBC, 2019. (Série Metodologia de Pesquisa em Informática na Educação, v. 2).

8.1 Future Work

There are several possibilities for future works which come to light from this thesis.

- Creation of new instruments to evaluate the CT skills.
- Psychometric validation of the theoretical model.
- Evaluation of items using IRT and other features, such as vocabulary, size of the text, number of elements, number of transformation for an element.
- Performing qualitative studies focused on the difficulty perception of the students.

- Investigation whether modifying the item can improve discrimination or change the difficulty level.

We have visualized some new research questions that can be addressed in future works.

- Is it possible to measure CT skills singly?
- How do CT skill is related to learning programming?
- How do computer science undergraduates develop their CT skill during the courses?
- Which are the further methods and metrics that can be used in order to assess CT skills?

All things considered, we have hoped this study arouse the interest of further investigation on CT.

Bibliography

[1] Ana Liz Souto O Araujo, Wilkerson Andrade, Dalton Guerrero, Monilly Melo, and Isabelle Maria Lima de Souza. Análise de rede na identificação de habilidades relacionadas ao pensamento computacional. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, page 655, 2018.

[2] Ana Liz Souto O Araujo, Wilkerson Andrade, Dalton Guerrero, Monilly Melo, and Isabelle Maria Lima de Souza. Como identificar habilidades do pensamento computacional? um estudo empregando analise fatorial. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 7, page 530, 2018.

[3] Ana Liz Souto O Araujo, Wilkerson L Andrade, Dalton D Serey Guerrero, and Monilly Ramos Araujo Melo. How many abilities can we measure in computational thinking?: A study on bebras challenge. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 545–551. ACM, 2019.

[4] Ana Liz Souto O Araujo, Jucelio Santos, Monilly Melo, Wilkerson Andrade, Dalton Guerrero, and Jorge Cesar Figueiredo. Teoria de resposta ao item. In Patrícia Augustin Jaques, Mariano Pimentel, Sean Siqueira, and Ig Bittencourt, editors, *Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa*, chapter 8. Porto Alegre, SBC, 2019.

[5] Ana Liz Souto O Araujo, Jucelio S Santos, Wilkerson L Andrade, Dalton D Serey Guerrero, and Valentina Dagienė. Exploring computational thinking assessment in introductory programming courses. In *2017 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE, 2017.

[6] Frank B Baker. *The basics of item response theory*. ERIC, 2001.

[7] Frank B Baker and Seock-Ho Kim. *The basics of item response theory using R*. Springer, 2017.

[8] Valerie Barr and Chris Stephenson. Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1):48–54, 2011.

[9] Ashok Basawapatna, Kyu Han Koh, Alexander Repenning, David C. Webb, and Krista Sekeres Marshall. Recognizing computational thinking patterns. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, pages 245–250, New York, NY, USA, 2011. ACM.

[10] Carlo Bellettini, Violetta Lonati, Dario Malchiodi, Mattia Monga, Anna Morpurgo, and Mauro Torelli. How challenging are bebras tasks?: an irt analysis based on the performance of italian students. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 27–32. ACM, 2015.

[11] Daphne Blokhuis, Peter Millican, Chris Roffey, Eljakim Schrijvers, and Sue Sentance. UK Bebras Computational Thinking Challenge. `http://www.bebras.uk/uploads/2/1/8/6/21861082/ukbebras2015-answers_1.pdf`, 2015. Online; accessed 29 January 2016.

[12] Kay-Dennis Boom, Matt Bower, Amaël Arguel, Jens Siemon, and Antonia Scholkmann. Relationship between computational thinking and a measure of intelligence as a general problem-solving ability. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 206–211. ACM, 2018.

[13] Denny Borsboom and Angélique OJ Cramer. Network analysis: an integrative approach to the structure of psychopathology. *Annual review of clinical psychology*, 9:91–121, 2013.

[14] Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the

development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, 2012.

[15] Timothy A Brown. *Confirmatory factor analysis for applied research.* Guilford Publications, 2014.

[16] Leonardo Dos Santos Costa, Ahemenson Cavalcante, Ana Liz Souto O Araujo, Wilkerson Andrade, and Dalton Guerrero. Um estudo exploratório da aplicação de pensamento computacional baseado nas perspectivas de professores do ensino médio. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 992, 2017.

[17] Mária Csernoch, Piroska Biró, János Máth, and Kálmán Abari. Testing algorithmic skills in traditional and non-traditional programming environments. *Informatics in Education*, 14(2):175–197, 2015.

[18] A Csizmadia, P Curzon, M Dorling, S Humphreys, T Ng, C Selby, and J Woollard. Computational thinking: A guide for teachers. *Google Scholar*, 2015.

[19] Paul Curzon, Mark Dorling, Thomas Ng, Cynthia Selby, and John Woollard. Developing computational thinking in the classroom: a framework. 2014. Available: https://pdfs.semanticscholar.org/e4f3/c24924c5a707a196b2015494c829c15618d1.pdf.

[20] Valentina Dagiene, Linda Mannila, Timo Poranen, Lennart Rolandsson, and Pär Söderhjelm. Students' performance on programming-related tasks in an informatics contest in finland, sweden and lithuania. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 153–158. ACM, 2014.

[21] Valentina Dagienė, Eimantas Pelikis, and Gabrielė Stupurienė. Introducing computational thinking through a contest on informatics: Problem-solving and gender issues. *Informacijos Mokslai/Information Sciences*, 73, 2015.

[22] Valentina Dagienė, Sue Sentance, and Gabrielė Stupurienė. Developing a two-dimensional categorization system for educational tasks in informatics. *Informatica*, 28(1):23–44, 2017.

[23] Valentina Dagienė, Gabriele Stupuriene, et al. Bebras-a sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education-An International Journal*, 15:25–44, 2016.

[24] Ana Liz Souto O de Araujo, Wilkerson L Andrade, and Dalton D Serey Guerrero. A systematic mapping study on assessing computational thinking abilities. In *Frontiers in Education Conference (FIE), 2016 IEEE*, pages 1–9. IEEE, 2016.

[25] Rafael Jaime De Ayala. *The theory and practice of item response theory*. Guilford Publications, 2013.

[26] Debra Dhillon. Predictive models of question difficulty: A critical review of the literature. In *Technical Report. AQA Centre for Education Research and Policy, Manchester, UK*, 2003.

[27] Vladimiras Dolgopolovas, Tatjana Jevsikova, Loreta Savulionienė, and Valentina Dagienė. On evaluation of computational thinking of software engineering novice students. In *Proceedings of the IFIP TC3 Working Conference "A New Culture of Learning: Computing and next Generations*, pages 90–99, 2015.

[28] Caitlin Duncan and Tim Bell. A pilot computer science and programming course for primary school students. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, pages 39–48. ACM, 2015.

[29] Fomentar el Pensamiento Computacional. Dr. scratch: automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 15(46), 2015.

[30] Sacha Epskamp, Angélique OJ Cramer, Lourens J Waldorp, Verena D Schmittmann, Denny Borsboom, et al. qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, 48(4):1–18, 2012.

[31] Sacha Epskamp and Eiko I Fried. A tutorial on regularized partial correlation networks. *Psychological methods*, 2018.

[32] Tereza Cristina Erthal. *Manual de psicometria*. Zahar, 1987.

[33] Michael W Eysenck and Mark T Keane. *Cognitive psychology: A student's handbook*. Psychology press, 2013.

[34] Sally Fincher. What are we doing when we teach programming? In *Frontiers in Education Conference, 1999. FIE'99. 29th Annual*, volume 1, pages 12A4–1. IEEE, 1999.

[35] Megean Garvin, Heather Killen, Jan Plane, and David Weintrop. Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 899–905. ACM, 2019.

[36] Claudia Hofheinz Giacomoni, Marcia de Lima Athayde, Cristian Zanon, and Lilian Milnitsky Stein. Teste do desempenho escolar: evidências de validade do subteste de escrita. *Psico-USF*, 20(1):133–140, 2015.

[37] Daniela Giordano and Francesco Maiorana. Use of cutting edge educational tools for an initial programming course. In *Global Engineering Education Conference (EDUCON), 2014 IEEE*, pages 556–563. IEEE, 2014.

[38] Lindsey Gouws, Karen Bradshaw, and Peter Wentworth. First year student performance in a test for computational thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, SAICSIT '13, pages 271–277, New York, NY, USA, 2013. ACM.

[39] Shuchi Grover, Stephen Cooper, and Roy Pea. Assessing computational learning in k-12. In *Proceedings of the 2014 Conference on Innovation ; Technology in Computer Science Education*, ITiCSE '14, pages 57–62, New York, NY, USA, 2014. ACM.

[40] Shuchi Grover and Roy Pea. Computational thinking in k–12 a review of the state of the field. *Educational Researcher*, 42(1):38–43, 2013.

[41] Joseph F Hair, William C Black, Barry J Babin, Rolph E Anderson, Ronald L Tatham, et al. *Multivariate data analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 1998.

[42] Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. *Fundamentals of item response theory*, volume 2. Sage, 1991.

[43] Lance Howarth, Peter Millican, Chris Roffey, and Sue Sentance. UK Bebras Computational Thinking Challenge. `http://www.bebras.uk/uploads/2/1/8/6/21861082/ukbebras2014-answers_1.pdf`, 2014. Online; accessed 29 January 2016.

[44] Chenglie Hu. Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 223–227. ACM, 2011.

[45] Li-tze Hu and Peter M Bentler. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural equation modeling: a multidisciplinary journal*, 6(1):1–55, 1999.

[46] Peter Hubwieser and Andreas Mühling. Investigating the psychometric structure of bebras contest: towards mesuring computational thinking skills. In *Learning and Teaching in Computing and Engineering (LaTiCE), 2015 International Conference on*, pages 62–69. IEEE, 2015.

[47] Claudio Simon Hutz, Denise Ruschel Bandeira, and Clarissa Marceli Trentini. *Psicometria*. Artmed Editora, 2015.

[48] Ulas Ilic, Halil İIbrahim Haseski, and Ufuk Tugtekin. Publication trends over 10 years of computational thinking research. *Contemporary Educational Technology*, 9(2):131–153, 2018.

[49] ISTE. Operational definition of computational thinking for k-12 education, 2011. "Available: http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2. Accessed in 2017-05-26".

[50] Filiz Kalelioglu, Yasemin Gülbahar, and Volkan Kukul. A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3):583, 2016.

[51] Cagin Kazimoglu, Mary Kiernan, Liz Bacon, and Lachlan MacKinnon. Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9:522–531, 2012.

[52] Kyu Han Koh, Hilarie Nickerson, Ashok Basawapatna, and Alexander Repenning. Early validation of computational thinking pattern analysis. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, pages 213–218, New York, NY, USA, 2014. ACM.

[53] Özgen Korkmaz, Recep Çakir, and M Yaşar Özden. A validity and reliability study of the computational thinking scales (cts). *Computers in Human Behavior*, 2017.

[54] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. Computational thinking for youth in practice. *Acm Inroads*, 2(1):32–37, 2011.

[55] James Lockwood and Aidan Mooney. Computational thinking in education: Where does it fit? a systematic literary review. *arXiv preprint arXiv:1703.07659. Available in: https://arxiv.org/pdf/1703.07659.pdf*, 2017.

[56] James Lockwood and Aidan Mooney. Developing a computational thinking test using bebras problems. 2018.

[57] Violetta Lonati, Dario Malchiodi, Mattia Monga, and Anna Morpurgo. Bebras as a teaching resource: classifying the tasks corpus using computational thinking skills. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pages 366–366. ACM, 2017.

[58] Violetta Lonati, Mattia Monga, Dario Malchiodi, and Anna Morpurgo. How presentation affects the difficulty of computational thinking tasks: an irt analysis. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, pages 60–69. ACM, 2017.

[59] Sze Yee Lye and Joyce Hwee Ling Koh. Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51–61, 2014.

[60] Linda Mannila, Valentina Dagiene, Barbara Demo, Natasa Grgurina, Claudio Mirolo, Lennart Rolandsson, and Amber Settle. Computational thinking in k-9 education. In

Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference, pages 1–29. ACM, 2014.

[61] Yoshiaki Matsuzawa, Kazuyoshi Murata, and Seiichi Tani. Multivocal challenge toward measuring computational thinking. In *Open Conference on Computers in Education*, pages 56–66. Springer, 2018.

[62] Jesús Moreno-León, Marcos Román-González, and Gregorio Robles. On computational thinking as a universal skill: A review of the latest research on this ability. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1684–1689. IEEE, 2018.

[63] Emiliano Oliveira and Ana Liz Araujo. Pensamento computacional e robótica: Um estudo sobre habilidades desenvolvidas em oficinas de robótica educacional. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 530, 2016.

[64] T. Palts, M. Pedaste, V. Vene, and L. Vinikienė. Tasks for assessing skills of computational thinking. In *ICERI2017 Proceedings*, 10th annual International Conference of Education, Research and Innovation, pages 2750–2759. IATED, 16-18 November, 2017 2017.

[65] Seymour Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.

[66] Seymour Papert and Idit Harel. Situating constructionism. *Constructionism*, 36(2):1–11, 1991.

[67] Luiz Pasquali. *Técnicas de exame psicológico–TEP: manual*, volume 23. São Paulo: Casa do Psicólogo, 2001.

[68] Luiz Pasquali. *Psicometria: teoria dos testes na psicologia e na educação*. Editora Vozes Limitada, 2017.

[69] Philippe Perrenoud. *Construir as competências desde a escola; trad*. Artmed Editora, 1999.

[70] Wolfgang Pohl and Hans-Werner Hein. Aspects of quality in the presentation of informatics challenge tasks. In *The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives—ISSEP 2015.*, page 21, 2015.

[71] Ricardo Primi, Acácia A Angeli Santos, Claudette Medeiros Vendramini, Fernanda Taxa, Franz August Muller, M de F Lukjanenko, and Isabel Silva Sampaio. Competências e habilidades cognitivas: diferentes definições dos mesmos construtos. *Psicologia: teoria e pesquisa*, 17(2):151–159, 2001.

[72] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[73] William Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois, 2017. R package version 1.7.8.

[74] Andrew Reynolds, Jim Mann, John Cummings, Nicola Winter, Evelyn Mete, and Lisa Te Morenga. Carbohydrate quality and human health: a series of systematic reviews and meta-analyses. *The Lancet*, 393(10170):434–445, 2019.

[75] Brandon Rodriguez, Stephen Kennicutt, Cyndi Rader, and Tracy Camp. Assessing computational thinking in cs unplugged activities. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 501–506. ACM, 2017.

[76] M Román-González. Computational thinking test: Design guidelines and content validation. In *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)*, pages 2436–2444, 2015.

[77] Marcos Román-González, Jesús Moreno-León, and Gregorio Robles. Complementary tools for computational thinking assessment. In *Proceedings of International Conference on Computational Thinking Education (CTE 2017), S. C Kong, J Sheldon, and K. Y Li (Eds.). The Education University of Hong Kong*, pages 154–159, 2017.

[78] Marcos Román-González, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández. Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 2016.

[79] Yves Rosseel. lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2):1–36, 2012.

[80] Linda Seiter and Brendan Foreman. Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 59–66. ACM, 2013.

[81] Cynthia Selby and John Woollard. Computational thinking: the developing definition. *University of Southampton (E-prints). Available in:* `https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf`, 2013.

[82] Cynthia C. Selby. Relationships: Computational thinking, pedagogy of programming, and bloom's taxonomy. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, WiPSCE '15, pages 80–87, New York, NY, USA, 2015. ACM.

[83] Cynthia Collins Selby. *How can the teaching of programming be used to enhance computational thinking skills?* PhD thesis, University of Southampton, 2014. 325pp. Available: https://eprints.soton.ac.uk/366256/.

[84] Mark Sherman and Fred Martin. The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, 30(6):53–59, 2015.

[85] W Van der Vegt. Predicting the difficulty level of a bebras task. *Olympiads in Informatics*, 7:132–139, 2013.

[86] W Van der Vegt. How hard will this task be? developments in analyzing and predicting question difficulty in the bebras challenge. *Olympiads in Informatics*, 12:119–132, 2018.

[87] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

[88] Jeannette M Wing. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725, 2008.

[89] Aman Yadav, Christina Krist, Jon Good, and Elisa Nadire Caeli. Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4):371–400, 2018.

[90] Aman Yadav, Chris Mayfield, Ninger Zhou, Susanne Hambrusch, and John T Korb. Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1):5, 2014.

[91] Ekaterina Yagunova, Sergey Podznyakov, Nina Ryzhova, Evgenia Razumovskaia, and Nikolay Korovkin. Tasks classification and age differences in task perception. case study of international on-line competition "beaver". In *Proc. of the 8th ISSEP Conf*, pages 33–43, 2015.

[92] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4):562–590, 2016.

[93] Iris Zur-Bargury, Bazil Pârv, and Dvir Lanzberg. A nationwide exam as a tool for improving a new curriculum. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 267–272, New York, NY, USA, 2013. ACM.

Appendix A

Termo de Consentimento Livre e Esclarecido

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
LABORATÓRIO DE PRÁTICAS DE SOFTWARE
Rua.: Aprígio Veloso, nº 882, Bairro Universitário, Campina Grande, PB.
CEP.: 58429-900 - Tel.: 2101-1429

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Você está sendo convidado(a) a participar como voluntário de uma pesquisa cujo título é "Avaliação de habilidades cognitivas no contexto de pensamento computacional" e que está sob a responsabilidade da pesquisadora Ana Liz Souto Oliveira de Araújo, aluna de doutorado do programa de Pós-Graduação em Computação da UFCG.

Após ser esclarecido(a) sobre as informações a seguir, no caso de aceitar fazer parte do estudo, assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Em caso de recusa, você não será penalizado(a).

O objetivo da pesquisa é analisar as habilidades cognitivas envolvidas no processo de resolução de problemas no contexto de pensamento computacional ao longo do curso de Ciência da Computação e suas relações com o aprendizado de programação. Os resultados contribuirão para identificar os processos cognitivos envolvidos na aprendizagem de programação ao longo do curso de Ciências da computação. A pesquisa está sendo realizada no contexto de três disciplinas de programação em três períodos distintos do curso de Ciência da Computação – UFCG. Todas as informações fornecidas serão tratadas de forma sigilosa e utilizadas na análise desta pesquisa. Os riscos envolvidos nesta pesquisa incluem você se sentir desconfortável em responder questões do teste. Caso isso ocorra, você pode parar de responder e receberá auxilio. Os benefícios desta pesquisa envolve a identificação de habilidades do pensamento computacional envolvidas no processo de aprendizagem em programação e, por consequência, podem levantar indícios que propiciem práticas pedagógicas que estimulem tais habilidades, bem como intervenções pedagógicas particulares aos alunos com dificuldades de aprendizagem de programação. Os resultados serão publicados em um relatório técnico.

O contato com a pesquisadora pode ser realizado por email analiz@copin.ufcg.edu.br ou no endereço Rua Aprígio Veloso, 882 - Universitário, Campina Grande – PB. Caso a pesquisa não esteja sendo realizada da forma esperada ou que prejudique o sujeito de alguma forma, você pode entrar em contato com o Comitê de Ética em Pesquisa do Hospital Universitário Alcides Carneiro - CEP-HUAC pelo telefone (83) 2101 - 5545 entre segunda e sexta-feira das 07h00 às 17h00 ou pelo email cep@huac.ufcg.edu.br.

CONSENTIMENTO DE PARTICIPAÇÃO DO SUJEITO

Concordo em participar do presente estudo como sujeito. Fui devidamente informado(a) e esclarecido(a) sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido que posso retirar meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade ou interrupção de meu acompanhamento.

Dados do participante da pesquisa	
Nome:	
RG:	CPF:
Telefone:	Email:

Campina Grande, _____ de _____________ de _____.

____
Assinatura do participante da pesquisa

____
Assinatura da pesquisadora

Appendix B

Instrument One

Nome completo: ____

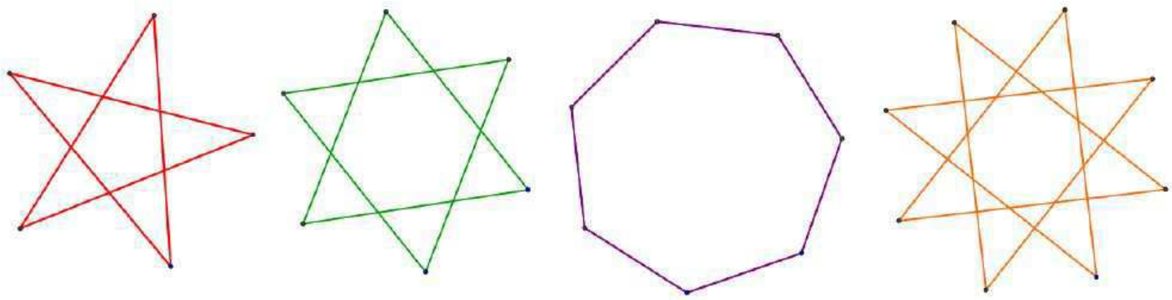
Matrícula: _______________________________________

1- Desenhando estrelas

Stella adora desenhar estrelas. Ela criou um sistema para marcar suas estrelas de acordo com a sua forma. Ela usa dois números:

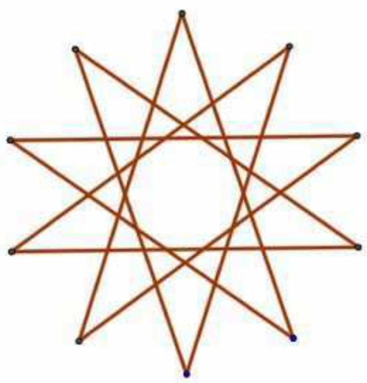
- O primeiro número refere-se a quantidade de pontos da estrela.
- O segundo número refere-se a um valor que indica se há uma linha reta, a partir de um ponto, para o ponto mais próximo (nesse caso o valor é 1), para o segundo ponto mais próximo (nesse caso o valor é 2), etc.

Abaixo estão quatro exemplos do sistema de marcação de Stella:



5:2 **6:2** **7:1** **8:3**

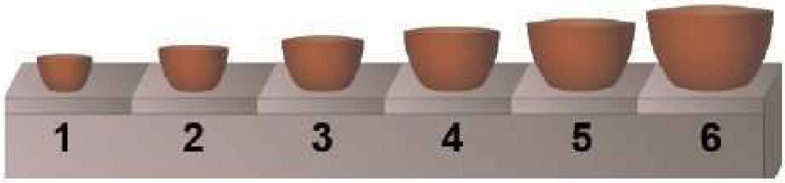
Questão: Como Stella deve marcar a estrela a seguir?



a) 9:3
b) 9:4
c) 10:4
d) 10:5

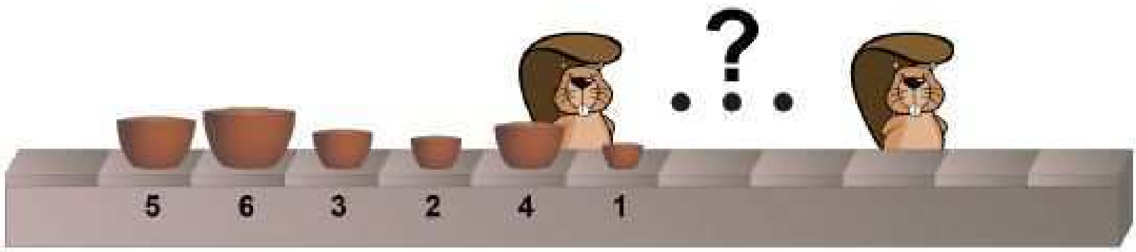
2- Fábrica de potes

Uma fábrica produz conjuntos de 6 potes de tamanhos diferentes. Uma longa esteira transportadora move os potes um por um, da esquerda para a direita.
O conjunto de 6 potes é colocado sobre a esteira transportadora em uma ordem aleatória.
Antes de embalar os potes, eles precisam ser ordenados da forma como aparece na figura abaixo:



Para ajudar na ordenação, a fábrica coloca os trabalhadores ao longo da esteira transportadora.
Quando um conjunto de potes passa por um trabalhador, ele deve trocar quaisquer dois potes vizinhos que estejam na ordem errada. O trabalhador vai continuar fazendo isso até que o conjunto de 6 potes tenha passado completamente pela esteira.
Veja como a ordem dos potes muda ao passar pelo trabalhador:

Questão: Quantos trabalhadores devem ser colocados ao longo da esteira transportadora para ordenar o conjunto de potes a seguir?________________________



3 - E-mail

Edgar está procurando uma nova casa.
Ele procurou na Internet e encontrou um apartamento perfeito por um preço muito bom.
Ele enviou um e-mail para Francisco, que está vendendo o apartamento, e recebeu uma resposta rápida:

```
Oi,
Obrigado pelo seu interesse no meu apartamento.
Embora eu não esteja na cidade, eu posso enviar-lhe a chave do apartamento para
que você possa visitá-lo, mas eu preciso de um depósito de segurança de
R$5.000,00 antecipado.

Para mostrar a minha confiança, eu anexei uma cópia do meu RG.
Abraço,
Francisco
```

Edgar não sabe o que fazer e está pedindo sua ajuda.

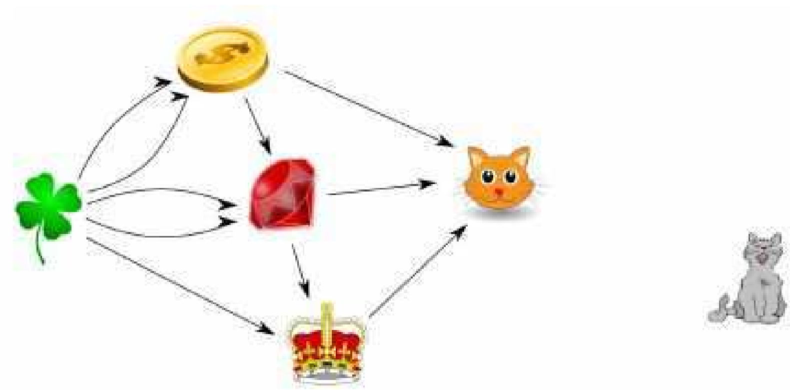
Questão: Qual o melhor conselho que você pode dar ao Edgar?

a) Pague o depósito. Com o RG você pode ir à polícia se você não receber o depósito de volta.
b) Isso é perfeito. Se você gostar do apartamento, você já pode ficar com a chave.
c) Não pague o depósito, há uma grande chance de ser uma fraude.
d) Pague o depósito, vá e veja o apartamento e decida-se mais tarde.

4 - Castor Alquimista

O Castor Alquimista pode converter objetos em novos objetos. Ele pode converter:

- Dois trevos em uma moeda;
- Uma moeda e dois trevos em um rubi;
- Um rubi e um trevo em uma coroa;
- Uma moeda, um rubi e uma coroa em um gatinho.



Depois que um objeto foi convertido em um outro objeto, ele desaparece imediatamente.

Questão: De quantos trevos o Castor Alquimista precisa para criar um gatinho?

a) 5 b) 10 c) 11 d) 12

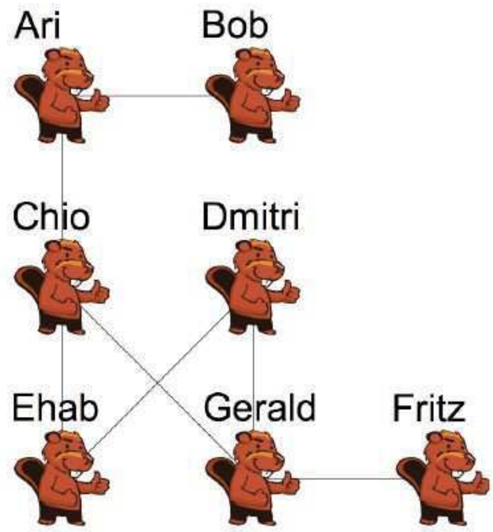
5 - Tutorial

O professor da escola de castores deseja distribuir um material para seus alunos.
Ele encontrou um portal com um livro digitalizado que declara em sua primeira página que ele deve ser distribuído de acordo com a "*Creative Commons License*" (CC-BY-ND) que concede liberdade para todo mundo compartilhar, copiar e distribuir o material em qualquer meio ou formato para qualquer finalidade, mesmo comercialmente, desde que seja dado o crédito apropriado ao autor.
A licença também especifica que, se forem feitas alterações, traduções ou adições, o livro modificado não pode ser distribuído.

Questão: Qual dessas ações não é permitida sob os termos dessa licença?
a) Vender cópias do livro para os alunos.
b) Tradução do livro, mantendo a cópia traduzida para si mesmo.
c) Dar aos alunos um capítulo de sua tradução do livro.
d) Colocar uma cópia do livro no site da escola.

6 - Popularidade

Sete castores estão em uma rede social online chamada Instadam. A rede social Instadam só permite que eles vejam as suas próprias fotos e as das páginas de seus amigos.
No diagrama abaixo, se dois castores são amigos, eles estão ligados por uma linha.
Após as férias de Verão, todos postam uma foto de si mesmo em todas as páginas de seus amigos.

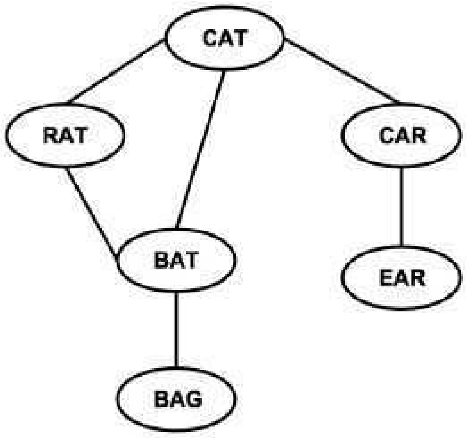


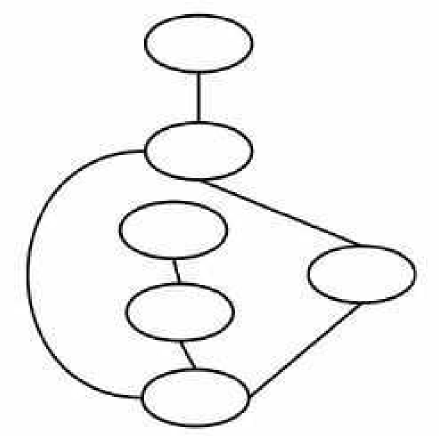
Questão: A foto de qual castor será a mais vista?

a) Ari
b) Chio
c) Dmitri
d) Ehab
e) Gerald

7 - Troca de palavras

Na sua lição de casa, Tomás teve que escrever palavras em inglês nos cartões e conectá-los com elásticos.
O professor de inglês pediu para ligar quaisquer duas palavras que se diferenciam em exatamente uma letra.
Tomás fez isso, como você pode ver na imagem à direita.



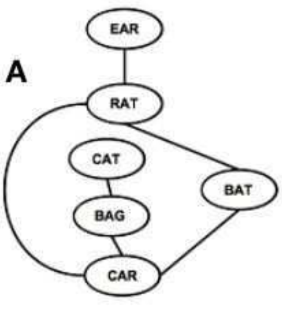


Tomás parou de estudar por um tempo, e quando voltou, ele teve uma surpresa.
Pedro, seu irmão mais novo, tinha apagado todas as palavras!
Além disso, os cartões foram completamente misturados, como você pode ver na imagem à esquerda.

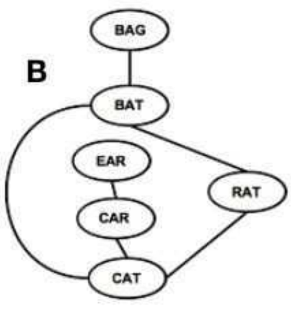
É importante ressaltar que as linhas com elástico ainda estavam conectadas como antes.
Tomás tem certeza que ele poderia colocar as palavras de volta no lugar correto.

Questão: Qual das respostas contêm as palavras nos lugares corretos?

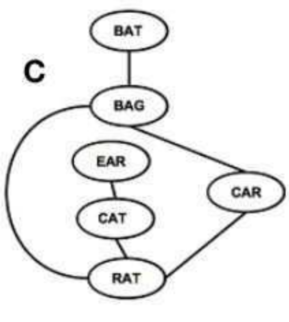
A



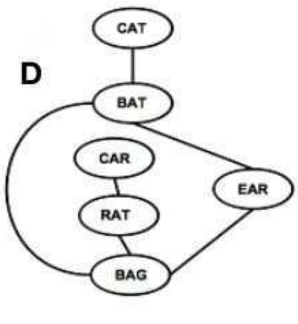
B



C



D

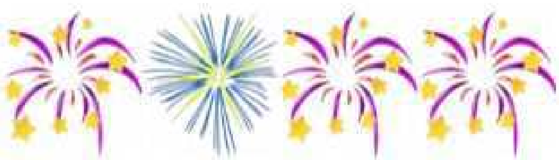


8 - Fogos de artifício

Dois castores vivem em tocas separadas por uma grande floresta.
Eles decidiram enviar mensagens entre si, lançando fogos de artifício.
Cada mensagem é uma sequência de palavras, embora os castores só conheçam cinco palavras diferentes.
Os castores podem lançar dois tipos de fogos de artifício, um após o outro, e conhecem os seguintes códigos:

Word	Code
Tronco	
Árvore	
Pedra	
Rio	
Comida	

Por exemplo, para mandar a mensagem (bastante estranha) "Comida, Tronco, Comida", o castor deveria lançar:

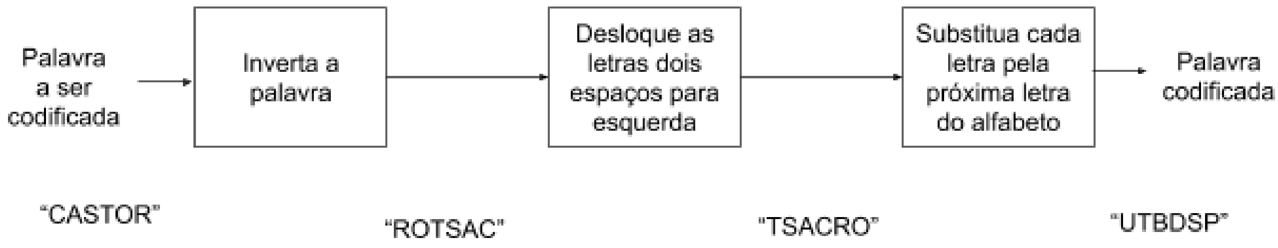


Questão: Quantos significados <u>diferentes</u> a mensagem a seguir possui?

a) 1 b) 2 c) 3 d) 4

9 - Você não irá descobrir

Os castores Alex e Betty enviaram mensagens um ao outro com a seguinte sequência de transformações em cada palavra.



Por exemplo, a palavra “CASTOR” foi transformada em “UTBDSP”
Betty recebeu a seguinte mensagem “BWBDPM” de Alex.

Questão: O que Alex queria dizer?

a) GAIATO b) CASUAL c) CAVALO d) PEIXES

10 - Vire as cartas

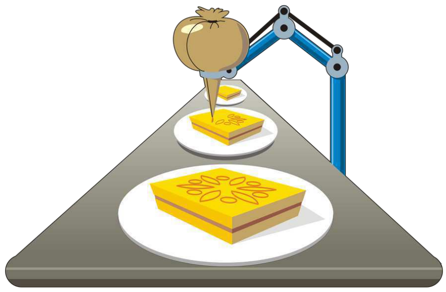
Chris mostrou suas quatro cartas:

Cada carta tem uma letra em um lado e um número no lado oposto.
Chris disse: "Se houver uma vogal em um lado da carta, então há um número par no lado oposto da mesma carta."

Questão:

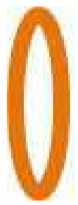
Escreva quais são as cartas que você precisa virar para comprovar se a afirmação de Chris está correta?

11- Decorando com chocolate

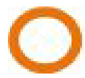


Tudo é automatizado na fábrica de chocolate: os doces estão deslizando em uma esteira transportadora, e há um robô com uma seringa que desenha formas diferentes com chocolate.
O robô executa os seguintes comandos:

`Folha` – desenha:



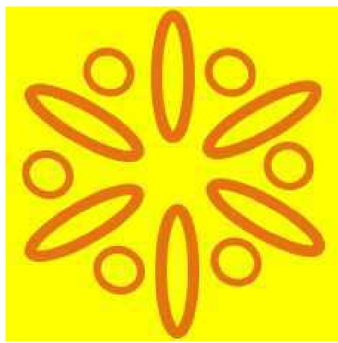
`Círculo` – desenha :



`Rotacione k` –
rotaciona o doce em sentido horário em k°.

`Repita n`
`[…`
`]`
– repete o comando dentro do colchetes n vezes

Para decorar o doce com flores, foram escritos algumas sequências para o robô executar.

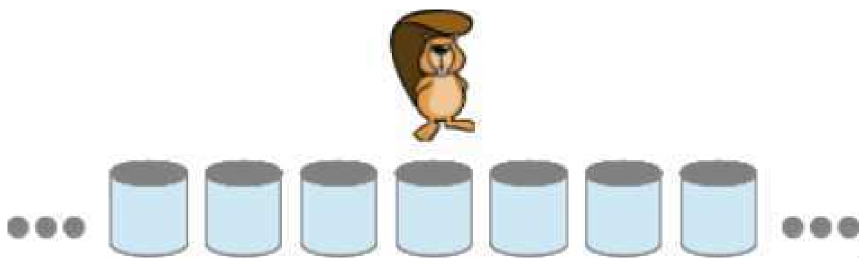


Questão:
Qual das sequências de comandos abaixo o robô <u>NÃO</u> desenha a flor?

A.	B.	C.	D.
Repita 6 [Rotacione 30 Círculo Rotacione 30 Folha]	Repita 6 [Folha Rotacione 60] Rotacione 330 Repita 6 [Círculo Rotacione 300]	Repita 6 [Folha Rotacione 60] Repita 6 [Círculo Rotacione 60]	Repita 6 [Folha Rotacione 60] Rotacione 30 Repita 6 [Círculo Rotacione 60]

12 - Castor atarefado

O castor Gump está muito ansioso. Por isso, o castor Alan contratou-o para preencher seu estoque de alimentos. O armazenamento é feito em uma fila de cestos e cada cesto pode estar "vazio" ou "cheio". Inicialmente, os cestos estão vazios e o Gump está de frente para um dos cestos vazios.



Sempre que Gump estiver de frente para um cesto vazio, ele irá preenchê-lo com um alimento. Mas, para se mover ao longo da fileira de cestos, Alan deu a Gump um conjunto muito especial de instruções, das quais Gump deve combinar de acordo com a situação.

Selecionar as instruções, depende de: (1) se o espaço em frente a Gump está vazio ou cheio, e (2) o humor de Gump (que pode ser "tranquilo" ou "nervoso").

Uma instrução diz a Gump para (1) passar para a próxima cesta à esquerda ou à direita e (2) avaliar seu humor, se está tranquilo ou nervoso - ou PARAR todas as atividades. Assim, uma instrução pode ser, por exemplo, (esquerda, nervoso).

Alan deu as instruções na seguinte tabela:

	Tranquilo	**Nervoso**
Cesta vazia	(direita, nervoso)	(esquerda, tranquilo)
Cesta cheia	(esquerda, nervoso)	(direita, PARAR)

Questão:

Considerando que o humor inicial de Gump é "**Tranquilo**":

Escreva quantas cestas vazias serão preenchidas quando Gump **PARAR**?

Appendix C

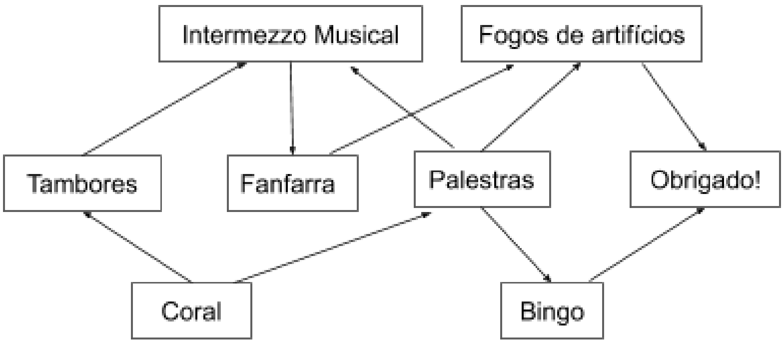
Instrument Two

Nome completo: ____

Cerimônia

Organizar um dia de festa é trabalhoso em Bebras City. Todos os eventos devem ocorrer em uma ordem específica.

O diagrama mostra todos os eventos que devem ser incluídos. As setas indicam que um evento deve ocorrer antes de um outro evento. Por exemplo, o Intermezzo Musical só pode acontecer depois de ambos, os Tambores e as Palestras, terem terminado.



Ordene os eventos do dia enumerando-os na sequência das regras mostradas no diagrama.

( ) Fogos de artifícios

( ) Bingo

( ) Obrigado!

( ) Palestras

( ) Tambores

( ) Coral

( ) Fanfarra

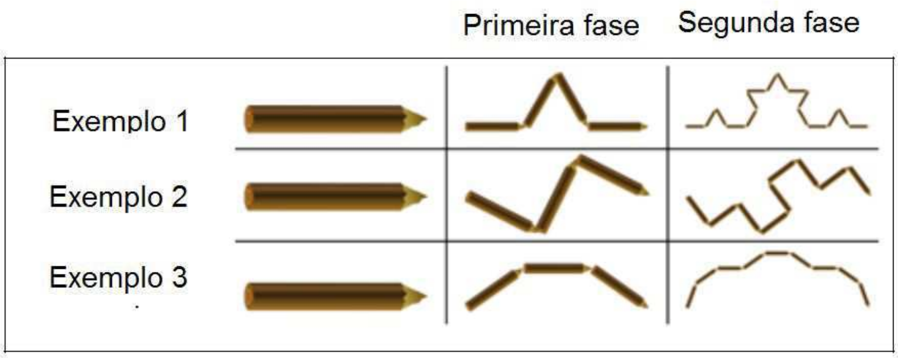
( ) Intermezzo Musical

Arte em toras

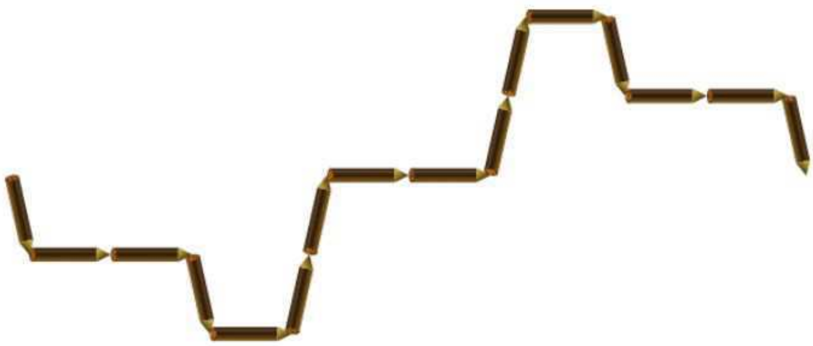
Quando castores roem árvores eles gostam de colocar as toras de uma maneira especial.

Os castores começam com uma única tora. Na primeira fase, uma tora grande é roída em toras menores. Na próxima fase, cada tora individual é novamente roída em toras ainda menores, mas sempre mantendo o padrão de início. Isso fica se repetindo.

Aqui estão três exemplos. Em cada linha você pode ver como o castor começou, o resultado após a primeira fase e o resultado após a segunda fase.



Se o resultado da segunda fase for:



Qual foi a primeira fase?

a) 

b) 

c) 

d) 

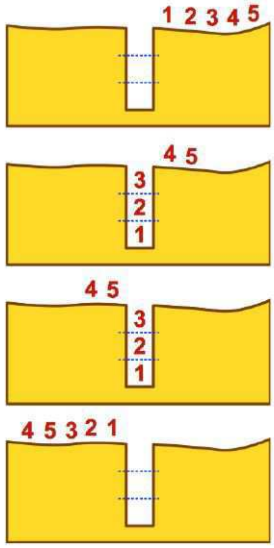
Castores em fuga

Uma colônia de castores está viajando por uma floresta escura. O caminho é estreito, então eles viajam em fila sem ultrapassagem.

Algumas vezes há um buraco no caminho. Um buraco é atravessado da seguinte maneira:

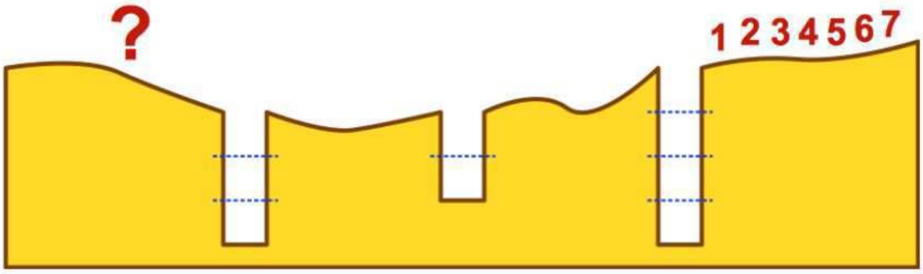
- Primeiro, os castores entram um a um no buraco, até preenchê-lo.
- Em seguida, toda a colônia passa pelo buraco.
- Os castores que entraram no buraco irão sair.

As imagens à direita mostram como cinco castores passaram por um pequeno buraco que foi preenchido por três castores.



Uma colônia de 7 castores está atravessando a floresta. Eles passaram por 3 buracos.

O primeiro buraco foi preenchido por 4 castores, o segundo por 2 e o último por 3 castores.



Qual a ordem em que os castores se encontram depois de passarem pelo terceiro buraco?

a) 4756123

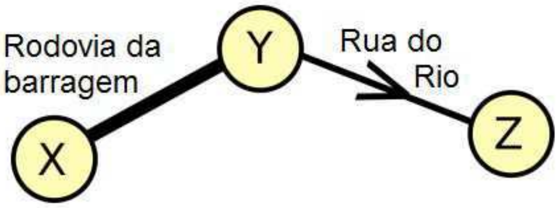
b) 6574321

c) 2165347

d) 5761432

Tráfego na vila

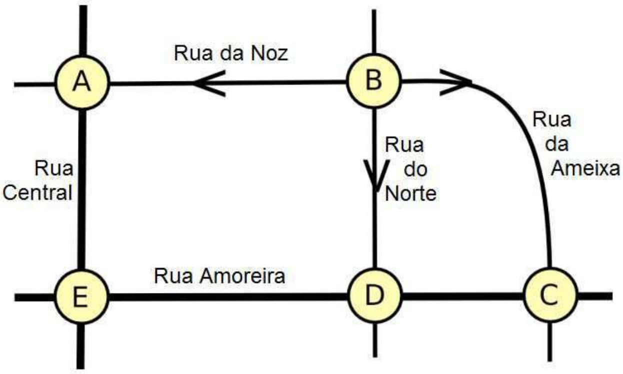
Em uma pequena vila existe uma rua de mão única e uma rua de mão dupla. A fim de ajudar o taxista da cidade, uma tabela foi criada para mostrar as rotas que podem ser feitas. Abaixo está o mapa e a tabela correspondente.



		PARA		
		X	**Y**	**Z**
DE	**X**		✓	
	Y	✓		✓
	Z			

Castor Vilage é uma vila maior e também precisa de uma tabela para os seus taxistas:

Preencha a tabela a seguir para os taxistas.

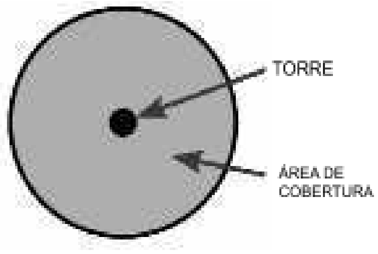


	A	B	C	D	E
A					
B					
C					
D					
E					

Rede à prova de tempestade

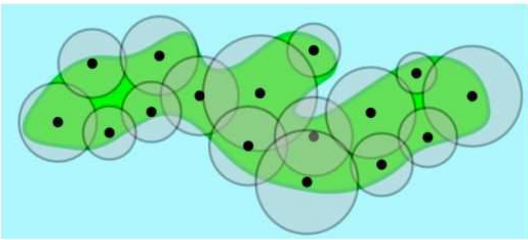
Em uma pequena ilha, uma rede de torres de celular é configurada. Cada torre abrange uma área circular da ilha.

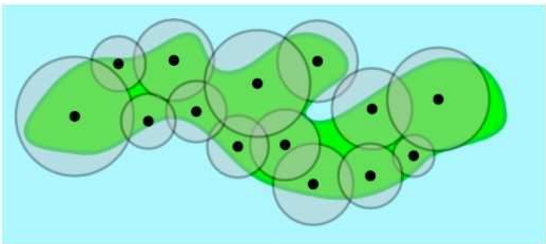
Quando a área de cobertura de duas torres se sobrepõe, as torres são chamadas de torres diretamente conectadas. As torres também podem ser indiretamente conectadas, se existe uma cadeia de torres diretamente conectadas entre as duas torres.

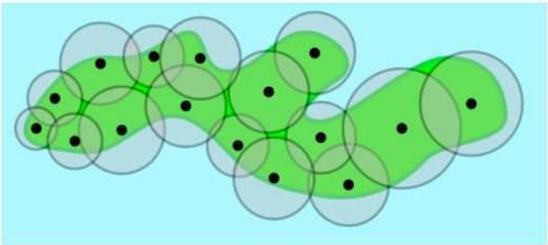


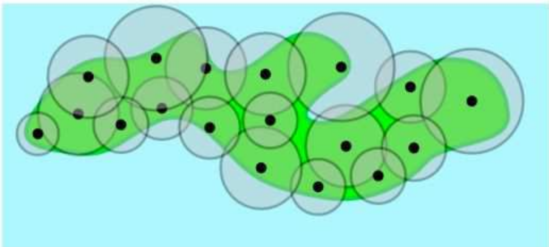
Os operadores querem fazer a rede de torres à prova de tempestade. Isto significa que, mesmo que uma torre pare de funcionar, todas as outras torres ainda devem estar conectadas, diretamente ou indiretamente.

Qual dos sistemas mostrados abaixo é uma maneira de criar uma rede à prova de tempestade na ilha?









Labirinto espacial

Alguns exploradores do espaço pousaram em um planeta vazio. Da sua nave podiam ver um labirinto com um objeto de prata desconhecido nele.

Os exploradores lançaram seu robô no labirinto esperando que ele pudesse dar uma olhada mais de perto no objeto desconhecido. Infelizmente o robô quebrou durante a queda e agora só pode enviar e receber comandos ilegíveis sobre onde ir.

O robô sugere quatro possíveis direções para seguir. Mesmo que as palavras das instruções estejam ilegíveis, ainda há apenas quatro palavras que são diferentes entre si, cada uma indicando norte, oeste, leste ou sul.

Ao seguir as instruções, o robô caminhará para um quadrado adjacente (próximo quadrado) conforme as instruções.

Que instruções os exploradores devem enviar ao robô para que ele chegue ao objeto de prata?

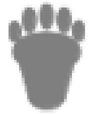
a) Ha' poS poS Ha' Ha' nIH

b) Ha' poS poS Ha' nIH Ha'

c) Ha' Ha' poS Ha'

d) Ha' poS nIH vI'ogh Ha' poS

Pegadas

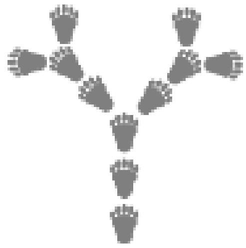


As instruções para a **árvore-1**:

- Siga em frente 1 passo para deixar uma pegada, volte na sua própria pegada.

Quando você souber como fazer uma **árvore-1**, você pode aprender como fazer uma **árvore-2**:

- Siga em frente 2 passos para deixar duas pegadas.
- Vire à esquerda e faça uma **árvore-1**.
- Vire à direita e faça uma **árvore-1**.
- Volte em suas próprias pegadas.

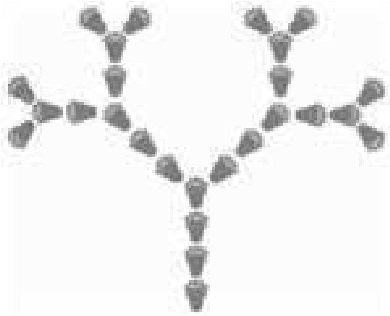


É fácil explicar como criar uma **árvore-3** porque uma **árvore-3** consiste em **árvores-2**:

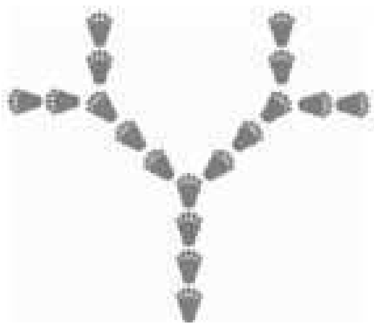
- Siga em frente 3 passos para deixar três pegadas.
- Vire à esquerda e faça uma **árvore-2**.
- Vire à direita e faça uma **árvore-2**.
- Volte em suas próprias pegadas.

De maneira semelhante, você pode criar uma **árvore-4**.

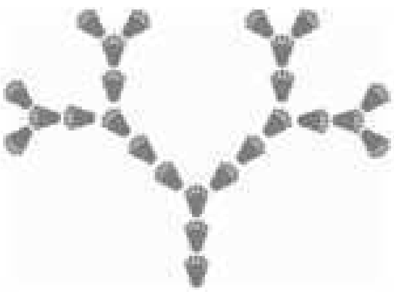
Qual das seguintes árvores é uma **árvore-4**?



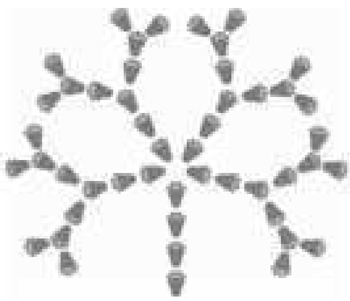
a)



b)



c)



d)

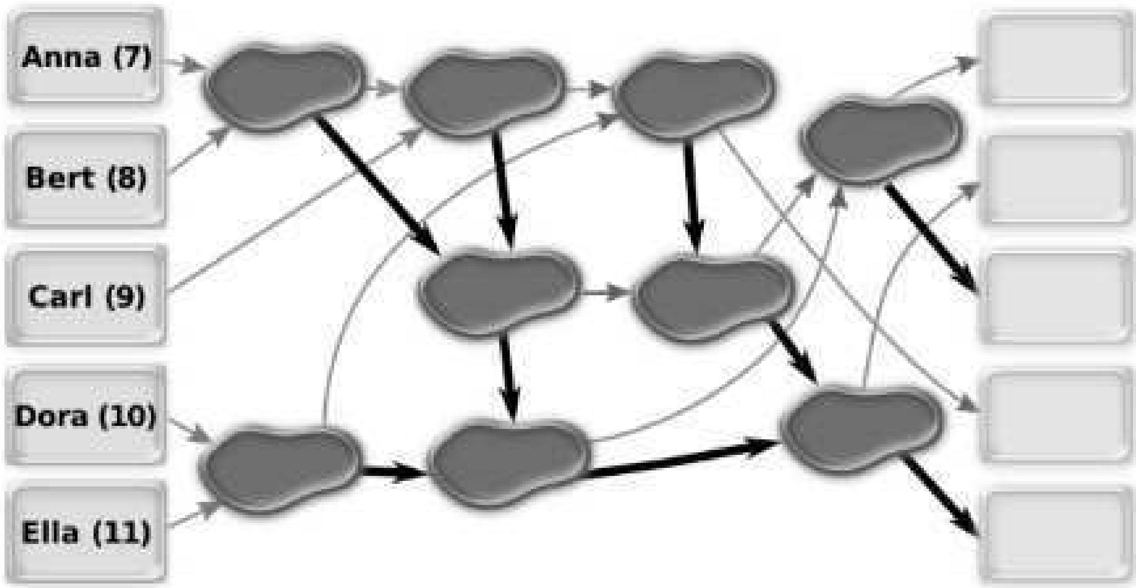
Saltos em poças d'água

Anna (7 anos), Bert (8 anos), Carl (9 anos), Dora (10 anos) e Ella (11 anos) estão participando de um jogo no qual eles pulam em poças d'água.

Eles colocaram setas entre as poças e todos eles começam no lado esquerdo, conforme indicado.

Quando uma criança salta em uma poça, ele ou ela espera até que uma segunda criança também chegue na poça. Quando isso acontecer, a criança mais velha pula seguindo a seta grossa e a mais nova segue a seta fina.

Escreva os nomes das crianças no lado direito do campo para mostrar onde cada criança termina.



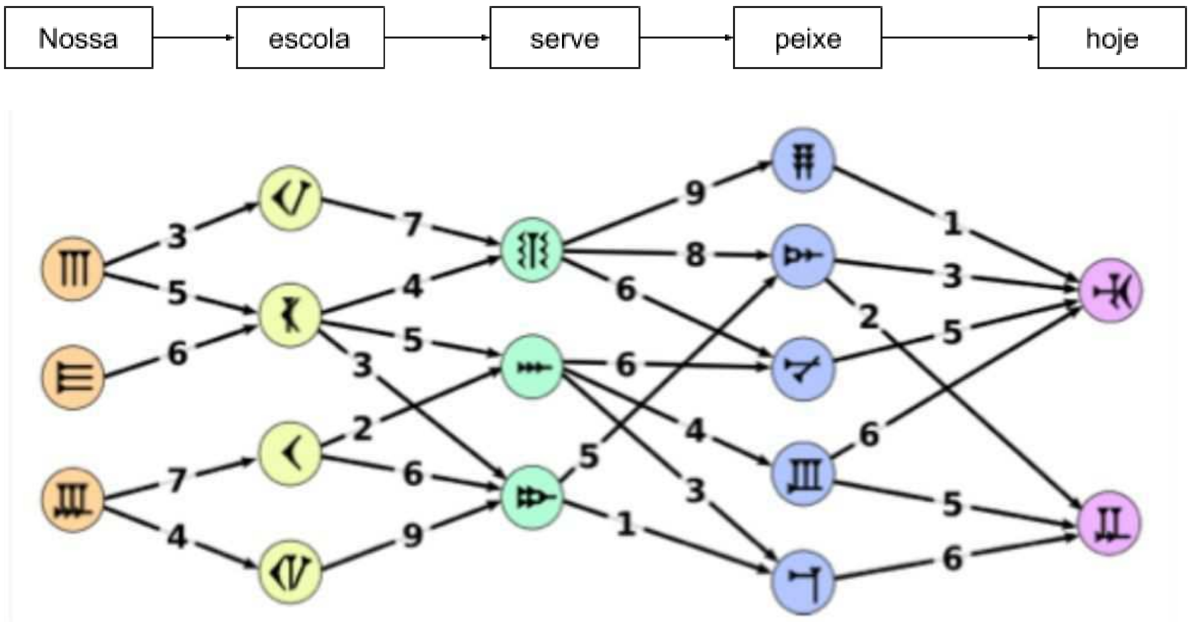
Melhor tradução

Aysha está tentando traduzir uma frase em português para uma língua antiga. Cada palavra traduz-se em um símbolo estrangeiro. Existem vários símbolos possíveis para cada palavra. Betty quer encontrar a melhor tradução.

Sob cada palavra em português Betty escreveu os possíveis símbolos estrangeiros. Entre cada par de símbolos, ela indicou o quão bem esses símbolos fazem sentido se usados nessa ordem. (Quanto mais alto o número, mais faz sentido utilizá-los).

A melhor tradução seria os cinco símbolos que produzem, quando somados, a maior pontuação possível.

Betty preparou a tradução de 'Nossa escola serve peixe hoje':

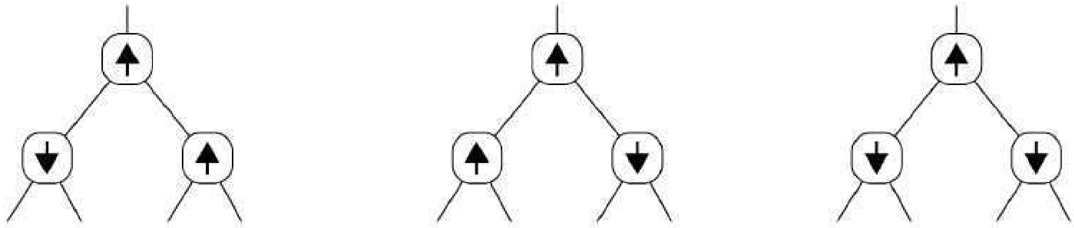


Marque com X os símbolos (um símbolo para cada palavra) que melhor traduzem a frase acima.

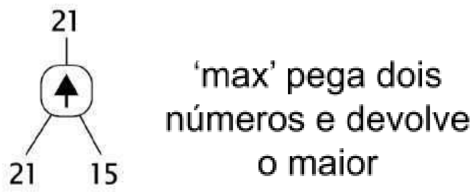
Máquina quebrada

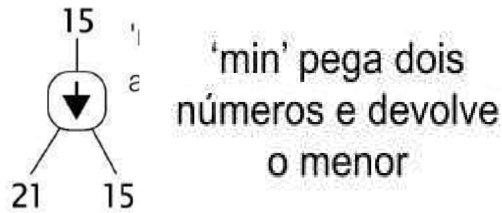
Hans construiu três máquinas, com a intenção de que elas produzissem como saída o segundo maior valor a partir de uma lista de quatro números.

Máquina 1 Máquina 2 Máquina 3



As máquinas podem usar dois componentes diferentes, chamados de 'max' e 'min'.





Em outras palavras, se os números representados por a, b, c e d são introduzidos em uma máquina nessa ordem, os resultados seriam como se segue:

Máquina 1: saída max(min(a,b), max(c,d))

Máquina 2: saída max(max(a,b), min(c,d))

Máquina 3: saída max(min(a,b), min(c,d))

Por exemplo, se Hilda insere os números 4, 3, 2, 1 na máquina 1, a saída que ela vai conseguir é 3, que é na verdade o segundo maior valor.

No entanto, durante o uso dos dispositivos, ela rapidamente percebeu que nenhuma das máquinas realmente funcionam. Na verdade, ela só precisava testar duas combinações de números para descobrir isso.

Quais das combinações abaixo ela usou para provar que nenhuma das máquinas funcionam?

a) 1, 2, 4, 3 e 2, 3, 4, 1

b) 2, 1, 3, 4 e 2, 3, 4, 1

c) 1, 4, 2, 3 e 2, 3, 4, 1

d) 1, 4, 2, 3 e 4, 1, 2, 3

Retângulos

Um robô foi programado para desenhar retângulos. Ele pode executar os seguintes comandos:

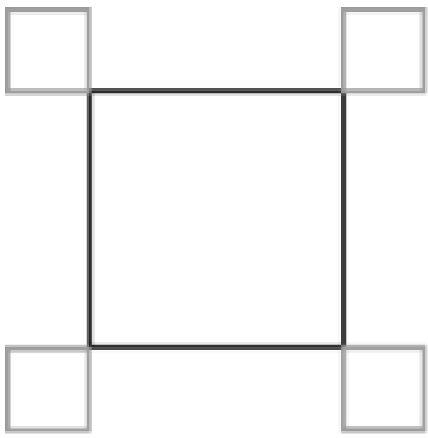
Cinza	desenhe uma linha cinza de tamanho 1
Preto	desenhe uma linha preta de tamanho 1
Vire	vire 90° no sentido horário

Além desses comandos simples, o robô também pode executar comandos complexos através da combinação de comandos.

Se A e B são comandos (simples e complexos) o robô pode executar:

A,B	primeiro execute o comando A e depois execute o comando B
n^x(B)	execute o comando B n-vezes

O robô deve desenhar a seguinte figura:



Qual das instruções abaixo **NÃO** resulta na figura acima?

a) 4^x(2^x(Cinza, Vire), Cinza, 3^x(Preto), Cinza, Vire)

b) 4^x(2^x(Cinza, Vire), 3^x(Preto), 2^x(Cinza, Vire))

c) 4^x(3^xPreto, 3^x(Cinza, Vire), Cinza)

d) 4^x(Preto, 3^x(Cinza, Vire), Cinza, 2^x(Preto))

Verdadeiro ou falso

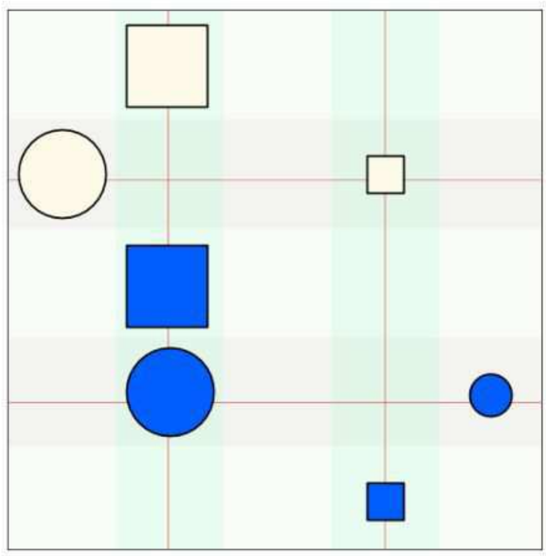
Alice e Tom estão jogando o jogo "Verdadeiro ou Falso" no quadro magnético em sua sala de aula.

Alice pregou sete peças magnéticas diferentes no quadro.

Ela, então, disse quatro declarações sobre a forma, a cor, o tamanho e a posição das peças.

Apenas uma declaração pode ser verdadeira e Tom deve descobrir qual delas é.

Qual dessas declarações abaixo é verdadeira?



a) Há duas peças X e Y, de modo que X é preto e Y é branca e X está acima de Y.

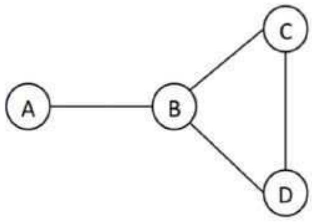
b) Para todos os pares de peças X e Y, se X é um quadrado e Y é um círculo, então X está acima de Y.

c) Para todos os pares de peças X e Y, se X é pequeno e Y é grande, então X está à direita de Y.

d) Para todos os pares de peças X e Y, se X é branca e Y é preto, então X está abaixo de Y.

Rede social

As imagens mostram as mesmas informações sobre as amizades entre os castores que vivem em uma toca.



	A	B	C	D
A		O		
B	O		O	O
C		O		O
D		O	O	

Por exemplo, castor A é amigo apenas do castor B (e o castor B também é amigo do castor A).

Se o castor A deseja tornar-se amigo do castor C, ele deve ser apresentado pelo castor B. O diagrama a seguir mostra as amizades entre 7 castores.

	A	B	C	D	E	F	G
A		O	O	O			
B	O		O	O			
C	O	O		O			
D	O	O	O		O		
E				O		O	O
F					O		O
G					O	O	

Qual é o número mínimo de apresentações que o castor A precisa para se tornar amigo do castor G?

a) 1

b) 2

c) 3

d) 4

Jogo da altura

Os jovens castores Ana, Bruno, Carol, Dani e Emanuel querem jogar com você.

Todos eles estão em uma fila. Em seguida, cada um deles conta quantos castores à frente e atrás são mais altos do que ele. Eles apresentaram os resultados em um pedaço de papel:

Nome	N° de castores mais altos	
	à frente	Atrás
Ana	1	2
Bruno	3	1
Carol	1	0
Dani	0	0
Emanuel	2	0

Em qual ordem os castores estão posicionados na fila?

a) Dani, Carol, Ana, Bruno, Emanuel

b) Ana, Carol, Dani, Emanuel, Bruno

c) Dani, Ana, Carol, Bruno, Emanuel

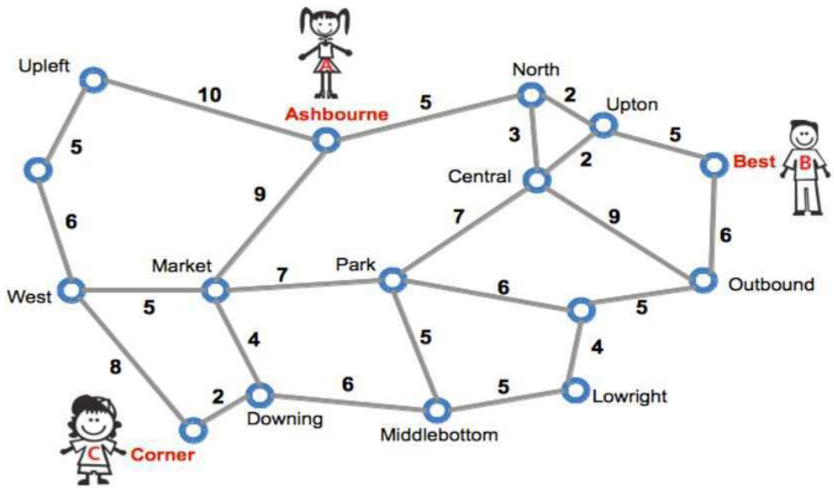
d) Dani, Ana, Emanuel, Bruno, Carol

Ponto de encontro

Três amigos Ana, Bruno e Clara vivem em uma cidade com um sistema de metrô.

O mapa do metrô abaixo mostra as estações e conexões entre as estações.

O mapa também indica quantos minutos cada conexão leva.



Ana mora ao lado da estação de Ashbourne, a estação mais próxima de Bruno é Best e a estação mais próxima de Clara é Corner.

Eles desejam selecionar uma estação para um encontro. Nenhum dos amigos deve demorar mais do que 15 minutos de viagem para chegar ao ponto de encontro.

Quais estações estão qualificadas como possíveis pontos de encontro?

Escreva abaixo **todas as estações** que se qualificam como pontos de encontro adequados.

Appendix D

Bebras Challenge 2015 from The United Kingdom

Drawing Stars

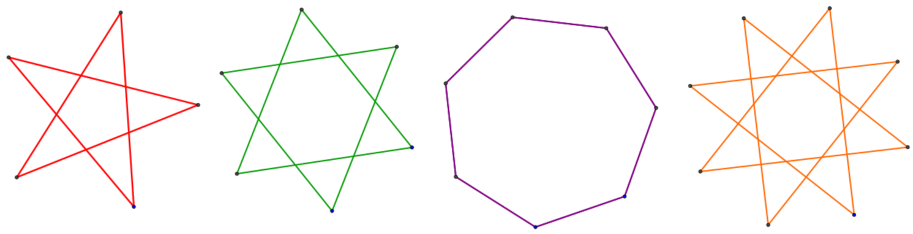
Kits:		Intermediates:	B
Castors:		Seniors:	
Juniors:		Elite:	A

Stella the beaver loves to draw stars. She has devised a system for labelling her stars according to their shape. She uses two numbers:

A number of dots for the star.

A number indicating if a line from a dot is drawn to the nearest dot (the number is 1), the second closest dot (the number is 2), etc.

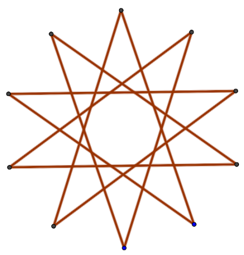
Here are four examples of Stella's labelling system:



5:2 **6:2** **7:1** **8:3**

Question:

How would Stella label the following star?

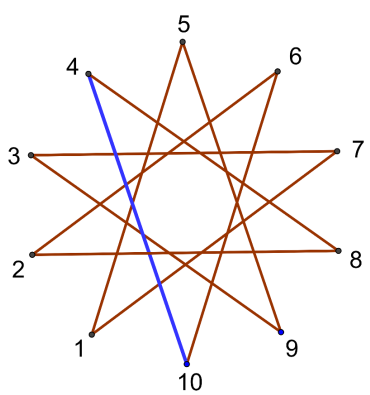


9:3 9:4 10:4 or 10:5

Answer:

10:4

Explanation:



It's Computational Thinking:

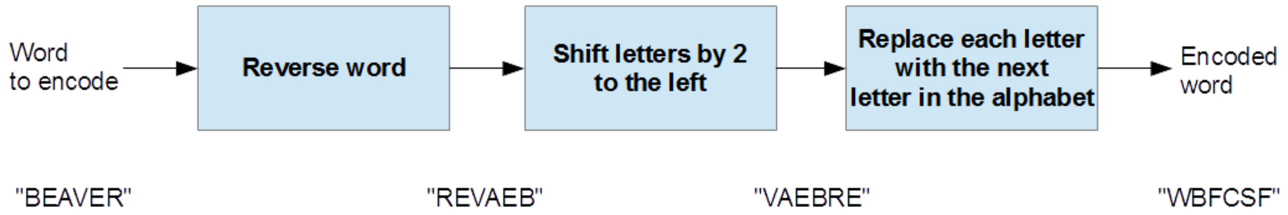
Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

Computers need simple representations of objects to be able to work with them. The fact that a complex and beautiful object such as a regular star polygon can be described by only two integers is an example of a simple representation.

You Won't Find It

Kits: Castors: Juniors: Intermediates: C Seniors: A Elite: A

Beaver Alex and beaver Betty send each other messages using the following sequence of transformations on every word.



For example, the word "BEAVER" is transformed to "WBFCSF".

Beaver Betty receives the encoded message "PMGEP" from beaver Alex.

Question:

What did Alex want to say?

RIVER, KNOCK, FLOOD or LODGE

Answer:

FLOOD

Explanation:

The steps of the transformation, applied in the reverse order, are:

“PMGEP” → “OLFDO” → “DOOLF” → “FLOOD” That is:

- Replace each letter with the previous in the alphabet;
- Shift letters by 2 to the right;
- Reverse word.

The other answers are not correct.

It’s Computational Thinking:

Skills - Algorithmic Thinking (AL), Evaluation (EV)

The image in this task is part of a simple flowchart, that explains how to change a word step by step. Flowcharts are a way to describe algorithms. In this task the algorithm changes text so that nobody can understand it. This is called ciphering.

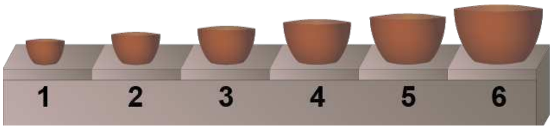
Bowl Factory

Kits:		Intermediates:	C
Castors:		Seniors:	C
Juniors:		Elite:	B

A factory produces sets of 6 bowls of different sizes. A long conveyor belt moves the bowls one by one, from left to right.

Bowl production places the 6 bowls of each set onto the conveyor belt in a random order.

Before packing the bowls, they need to be sorted to look like this:

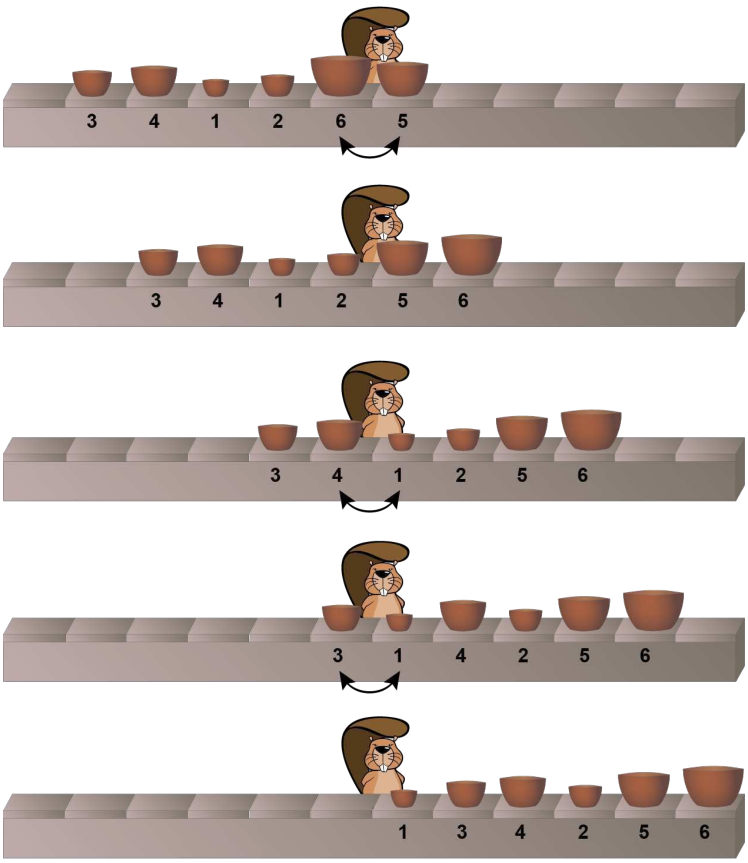


To help with the sorting, the factory places workers along the conveyor belt.

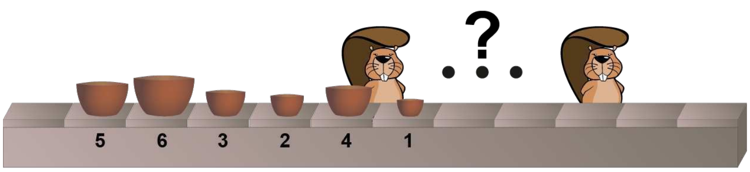
When a set of bowls passes a worker, the beaver will swap any two neighbouring bowls which are in the wrong order.

The worker will keep doing this until the set of 6 bowls has finished passing.

See how the order of a set of bowls changes as it passes one worker:



Question:
How many workers should be put along the line to sort the set of bowls on the right?



Answer:

4

Explanation:

As shown in the question, the original order of the set of bowls is: 5 6 3 2 1 4
Remember that the swapping of neighbouring bowls happens from right to left.

After passing a first worker, the order of the bowls is: 1 5 6 3 2 4 (4 swaps, all with bowl 1)
After passing a second worker, the order is: 1 2 5 6 3 4 (3 swaps, all with bowl 2)
After passing a third worker: 1 2 3 5 6 4 (2 swaps, all with bowl 3)
After passing a fourth worker, the set of bowls is sorted: 1 2 3 4 5 6 (2 swaps, all with bowl 4)

It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

Typically, automatic processing of data (which is what computational thinking is mostly about) is much easier when data is arranged according to some criteria – when it is sorted. Much effort has been spent by Computer Scientists on investigating sorting algorithms. The method for sorting sets of bowls that is described in this task is called a "bubble sort". This sorting algorithm steps through a list of objects again and again, swapping any neighbouring objects which are in the wrong order. The list is sorted when no swap occurs during a pass through the list.

Bubble sorting is quite easy to understand compared to other sorting algorithms. Unfortunately, it is not very efficient. For sorting 1000 items, a bubble sort may use up to half a million steps in the worst case. Better sorting algorithms would use only about 10 000 steps.

Email Scam

Kits:		Intermediates:	
Castors:		Seniors:	A
Juniors:		Elite:	A

Edgar is looking for a new home to live in.

He searched the internet and found a perfect flat for a very good price.

He has sent an e-mail to Francis, who is selling the flat, and received a quick reply:

> Hi,
>
> Thank you for your interest in my flat.
>
> Although I am not in town, I can send you the key to the flat so you can inspect it, but I need a security deposit of $5.000,- beforehand.
>
> To show my trustworthiness, I attach a copy of my ID.
>
> Cheers,
>
> Francis

Edgar is unsure what to do and is asking for your help.

Question:

What would be your best advice?

A. Pay the deposit. With the ID you can always go to the police if you don't get the deposit back.

B. That is perfect. If you like the flat, you can keep the key right away.

C. Don't pay the deposit, there is a high chance that this is a mail fraud.

D. Pay the deposit, go and have a look and decide later on.

Answer:

Response C

Explanation:

Response C would be the best advice. The copy of the ID could be "Photoshopped". You will not be able to meet the person to verify the ID. Statements 1 and 2 are not good as there is a high chance of not even receiving the key. Statement 4 is not good since the authenticity of the ID can't be proven.

It's Computational Thinking:

Skills - Evaluation (EV)

The Internet can be used to hide ones true identity and provides anonymity. Criminal people use this mechanism to get money from naive people. Very often spelling mistakes or high money values are within such emails and should raise awareness from the user.

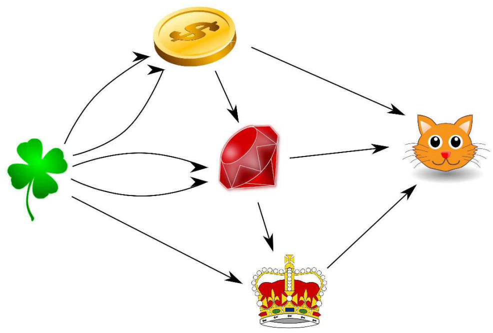
Beaver the Alchemist

Kits: Castors: Juniors: Intermediates: Seniors: C Elite: B

Beaver the Alchemist can convert objects into new objects. He can convert:

- Two clovers into a coin
- A coin and two clovers into a ruby
- A ruby and a clover into a crown
- A coin, a ruby, and a crown into a kitten.

After an object has been converted into another object, it disappears immediately.



Question

How many clovers does Beaver the Alchemist need to create one kitten?

5, 10, 11 or 12

Answer:

The answer is 11.

Explanation:

We can see the conversion as follows:

coin = 2 clovers
ruby = 2 clovers + 1 coin = 4 clovers
crown = 1 ruby + 1 clover = 4 clovers + 1 clovers = 5 clovers
kitten = 1 coin + 1 ruby + 1 crown = 2 clovers + 4 clovers + 5 clovers = 11 clovers

It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

This task demonstrates how graphs can be used to represent dependencies between items. A graph is a data structure that is used a lot in computational thinking to demonstrate relationships. Graphs also make it easier to visualise a task compared to just reading the descriptions of the relationships in text.

Popularity

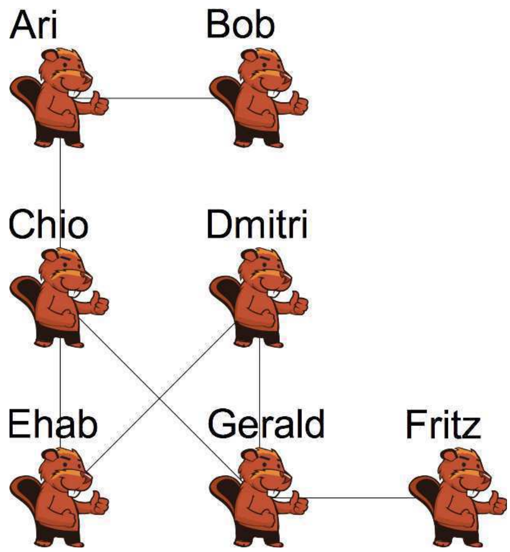
Kits:		Intermediates:	
Castors:		Seniors:	C
Juniors:		Elite:	B

Seven beavers are in an online social network called Instadam.

Instadam only allows them to see the photos on their own and their friends' pages.

In this diagram, if two beavers are friends they are joined by a line.

After the summer holidays everybody posts a picture of themselves on all of their friends' pages.



Question:

Which beavers' picture will be seen the most?

Ari, Bob, Chio, Dmitri, Ehab, Fritz or Gerald

Answer:

The correct answer is Chio.

Explanation:

In order to find the beaver whose picture gets seen by most beavers, you have to count the beavers that are at most two steps away. The beavers one step away are those on whose page the pictures will be posted and the beavers two steps away are those who can see these pages. Of course any beaver can only be counted once.

The following table summarises the info and helps us to see whose picture will be seen the most.

Beaver	Direct Friends	Friends' Friends	Total
Ari	Bob, Chio	Ehab, Gerald	4
Bob	Ari	Chio	2
Chio	Ari, Ehab, Gerald	Bob, Dmitri, Fritz	6
Dmitri	Ehab, Gerald	Chio, Fritz	4
Ehab	Chio, Dmitri	Ari, Gerald, Fritz	5
Fritz	Gerald	Chio, Dmitri	3
Gerald	Chio, Dmitri, Fritz	Ari, Ehab	5

It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

Many social networks use larger and more complicated versions of this concept. It is not always obvious that by posting something on a friend's page, it might be available to people other than the close friend.

Social networks themselves are incredibly powerful tools in today's world. Computing statistics on their users and their pages is useful to marketing departments and anyone else trying to understand a person or group of people.

Instadam could also be interpreted as a model of a miniature internet, with the beavers being websites and friends as pages "linked to". Search engines typically rank these websites by some measure of popularity or importance, at least by the number of links to and from the website.

A widely used way to find the result by using a computer is to use the flood fill algorithm which can cope with systems with more than the two iterations in this example..

Word Chains

Kits:		Intermediates:	
Castors:		Seniors:	C
Juniors:		Elite:	B

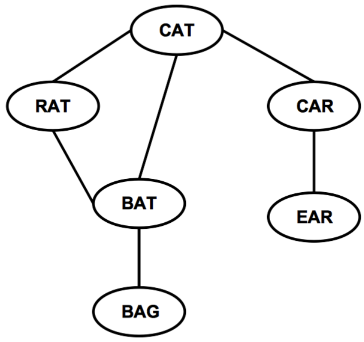
For his homework, Thomas had to write words on cards and connect them with rubber bands.

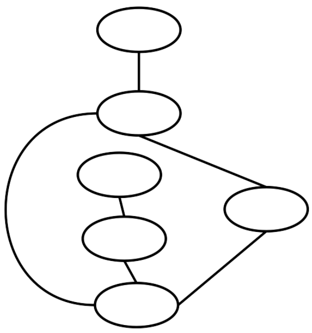
The teacher told him to connect any two words that differ by exactly one letter.

Thomas did this, as you can see in the picture on the right.

When Thomas returned from having a break he got a surprise.

Peter, his little brother, had erased all the words!





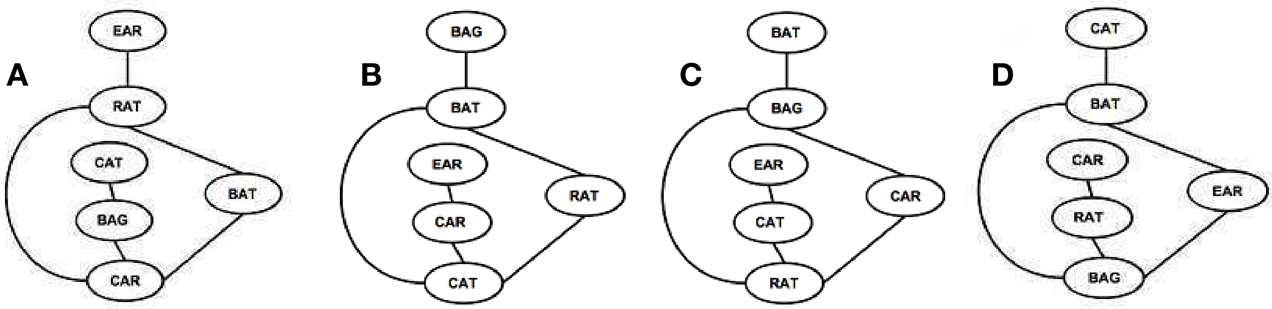
Also, the cards were completely mixed up, as you can see in the image on the left.

Importantly, the rubber bands still connected them as before.

Thomas was sure he could put the words back in the correct place.

Question:

Which of the pictures below contains the words in exactly the right places?



Answer:

B

Explanation:

We can proceed by counting the edges going from each node. There are 2 nodes with 3 edges, 2 nodes have two edges and 2 nodes have 1 edge. There is only one node with one edge connected to a node that has two edges. So we have identified the node for “EAR” and “CAR”. We can continue with this method ruling out the wrong answers as we proceed.

It’s Computational Thinking:

Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)

This is a problem about graphs. A graph is a set of objects, where some pairs of objects are connected

Fireworks

Kits:	Intermediates:	
Castors:	Seniors:	C
Juniors:	Elite:	

Two beavers live in lodges separated by a large forest.
They decide to send messages to each other by shooting fireworks into the sky above the trees.

Each message is a sequence of words, though the beavers only know five different words.
The beavers can shoot two types of fireworks, one after the other, and know the following codes:

Word	Code
Log	
Tree	
Rock	
River	
Food	

For example, to send the (rather strange) message "food, log, food", a beaver would shoot:

Question?

How many **different** meanings can the following sequence of fireworks have?

0, 1, 2, 3, or 4

Fireworks (hard)

Kits: Castors: Juniors: Intermediates: Seniors: Elite: C

In the Elite competition the same question was set but there was a free choice of input rather than a multi-choice question.

Answer:

The correct answer is 4.

Explanation:

The message could mean any of the following:

log, rock, food, river
log, log, log, river
rock, tree, river
rock,food, log, river

To convince yourself that there are no more possibilities, you can systematically count them:

- Start with the first firework. It is not a message, so you can write a zero on it.

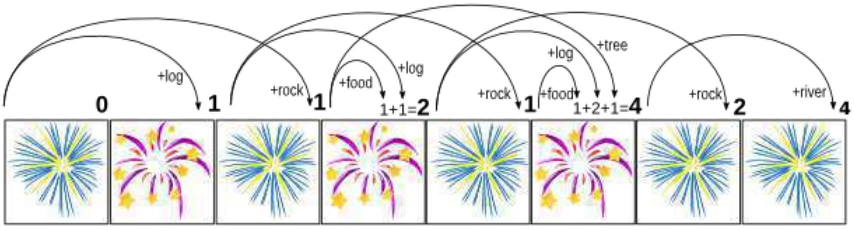
- The first two fireworks can only mean log. Write number one next to the second firework.

- We are at the third firework. It can have a meaning of any shorter sub-sequence plus one new word. Yet we see that there is no way to prolong the previously examined sequences (of length 1 and 2), so we only have one possible meaning (rock) and write 1 to the third firework.

- The fourth firework is finally somewhat interesting. It can either add the word log to the first two fireworks, or food to the first three fireworks, as shown by the arrows below. So we sum the two numbers at the 2nd and 3rd firework and write it to the 4th (1+1=2).

- We proceed applying the same idea to each firework to the right. We look one, two and three fireworks back. If those shorter messages can be prolonged with a correct word, we mark this fact with an arrow. Then we just sum the numbers "brought" by the arrows to the currently examined firework.

- At the last firework, we will have the number of all possible meanings.



It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

The process of building a solution systematically, step by step, using the previous steps like this is called dynamic programming. It makes the process much easier – just imagine trying to find all the full meanings of the message right away!

All digital information is represented using binary. That is, it consists of only the bits 0 and 1. Only longer combinations of 0 and 1 ("words" in this task) allow us to use more than two different meanings. But we also want to avoid ambiguity in our messages.

Most standard codes use the same number of bits per word, so there is only one meaning to each message. But if some word is used very often and some rarely, such code generates needlessly long messages. It is then useful to have shorter codes for frequent words (like "food") and longer codes for less frequent words (like "rock"). Of course you can be smarter than the beavers in our task: If you generate a prefix code the messages will only have one meaning. This trick of shortening frequent data chunks without introducing ambiguity is used in data compression.

The teacher in the beaver school wants to give some material to his students.

He found a portal with a scanned book which declares in its front page that it should be distributed according to a “Creative Commons License” (CC-BY-ND) that makes everyone free to share, copy and redistribute the material in any medium or format for any purpose, even commercially, provided that appropriate credit is given.

The license also specifies that if one remixes, translates, or builds upon the book, the modified book may not be distributed.

Question:

Which of these actions is not permitted under the terms of this license?

A. Selling copies of the book to students

B. Translating the book, keeping the translated copy for himself

C. Giving the students one chapter of his translation of the book

D. Putting a scanned copy of the book on the school website

Answer:

C is not permitted and so is the answer to this problem.

Explanation & It's Computational Thinking:

Skills - Evaluation (EV)

Copyright is a complex issue: sometimes even the experts in local and international laws have difficulties in disentangling regulations and people's rights. It is not easy to decide if something can be uploaded, downloaded, used, distributed. Creative Commons licenses were invented in order to make it easier for authors and users to understand what they can do without breaking laws or contracts. The authors can easily state clearly if they want attribution (a mention of the original author), if they allow commercial use of their creation, if they prohibit derivative works, and if they put restrictions on the license of derivative distributions. These “rights” (known, in order, as BY, NC, ND, SA) can be composed independently in a new license. So, for example, the task describes a BY-ND license, and users should immediately understand that the author requires attribution of her/his original work and no derivatives are permitted. Everything else is allowed (at least according to the license...). Thus:

- A is permitted by the author's license (but perhaps not by school regulations...),
- B is permitted, as long as the derivative work is not distributed to others,
- D is permitted, since Creative Commons do not restrict sharing if the attribution to the

original author is preserved.

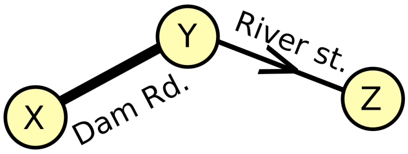
Appendix E

Bebras Challenge 2014 from The United Kingdom

Traffic in the city

Kits:		Intermediates:	B
Castors:		Seniors:	A
Juniors:		Elite:	A

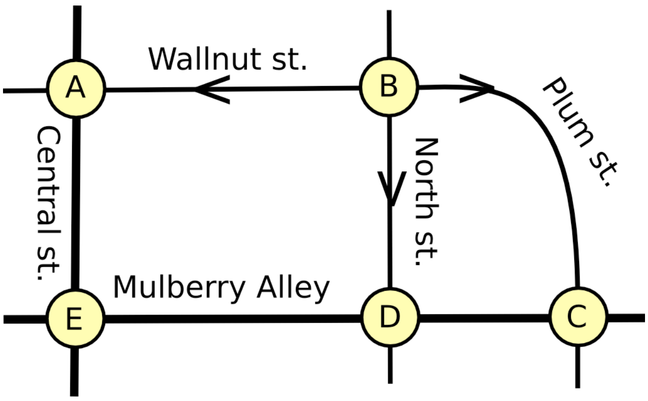
In a small village there is a one-way street and a two-way street. In order to help the village taxi driver a table is made to show which routes can be taken. Below is the map and the corresponding table.



		To		
		X	Y	Z
From	X		✓	
	Y	✓		✓
	Z			

Beaversville is a little larger and also wishes to have a table for its taxi drivers:

Fill in the table for the taxi drivers.



	A	B	C	D	E
A					
B					
C					
D					
E					

Answer:

	A	B	C	D	E
A					✔
B	✔		✔	✔	
C				✔	
D			✔		✔
E	✔			✔	

Explanation:

Remember you can only place a tick if the taxis can travel from the letter on the left to the letter at the top. For this to be possible there must be either an arrow facing in the correct direction or a line representing a two-way road between the two letters.

It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL)

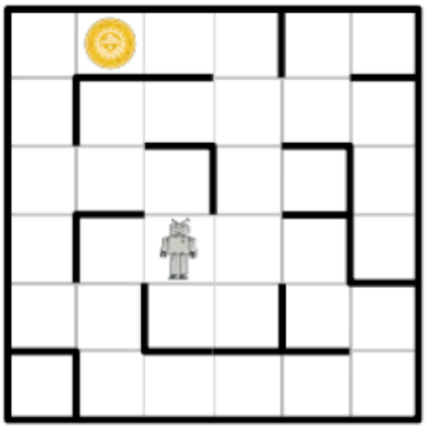
The table is known as an adjacency matrix. The table ignores the length and shape of all the streets and allows us to focus on whether or not there is a way to travel directly between two given intersections. So the table is an abstract model which ignores information that is not important to us. It is now much easier to store this information in a computer and more efficient to look up this information.

Space maze

Kits:	Intermediates:	
Castors:	Seniors:	A
Juniors:	Elite:	A

Some space explorers landed on an empty planet. From their ship they could see a maze with an unknown golden object in it.

The explorers dropped their robot into the maze hoping it could take a closer look at the unknown object. Unfortunately the robot broke during the fall and can now only send and receive garbled instruction about where to go.



The robot suggests four possible directions it can go. Even though the words in the instructions are garbled, there are still only four different words, each indicating north, west, east or south. When following the instructions the robot will move into an adjacent square as instructed.

Which instructions should the explorers send the robot in order for it to reach the golden object?

A. Ha' poS poS Ha' Ha' nIH

B. Ha' poS poS Ha' nIH Ha'

C. Ha' Ha' poS Ha'

D. Ha' poS nIH vI'ogh Ha' poS

Beavers on the run

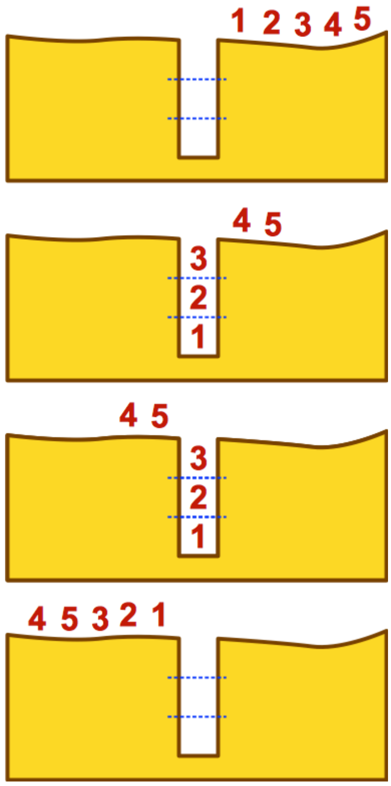
Kits:		Intermediates:	C
Castors:		Seniors:	B
Juniors:		Elite:	A

A colony of beavers is travelling through a dark forest. The path is narrow, so they travel in a row without passing.

Sometimes there is a hole in the path. A hole is passed in the following manner:

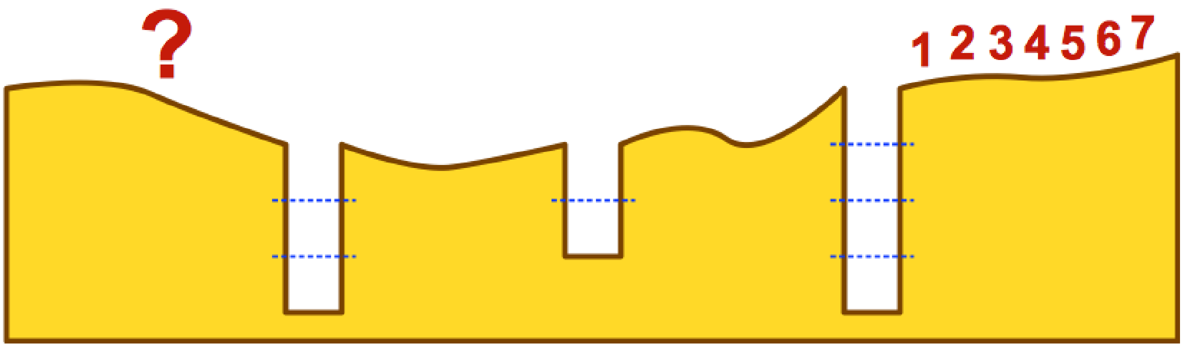
- First as many beavers jump into the hole as fit in.
- The entire colony will then pass across the hole.
- The beavers that jumped in will then climb out.

The images on the right show how five beavers pass a small hole that fits three beavers.



Question:

A colony of 7 beavers passed through the forest. They pass over 3 holes.
The first hole fits 4 beavers, the second fits 2, and in the last hole fit 3 beavers.



What order do the beavers find themselves after they have passed the third hole?

4756123

6574321

2165347

5761432

Answer:

2165347

Explanation:

Initially the line is 1 2 3 4 5 6 7

Then after the first hole of depth 4 we have:
5 6 7 4 3 2 1

After the second hole (depth 2) we have:
7 4 3 2 1 6 5

After the third hole (depth 3) we get:
2 1 6 5 3 4 7

It's Computational Thinking:

Concepts - Abstraction(AB), Decomposition(DE), Algorithmic Thinking(AL)

Organising data in a structured way is important in computer science and there are many different data structures that can be used for this purpose.

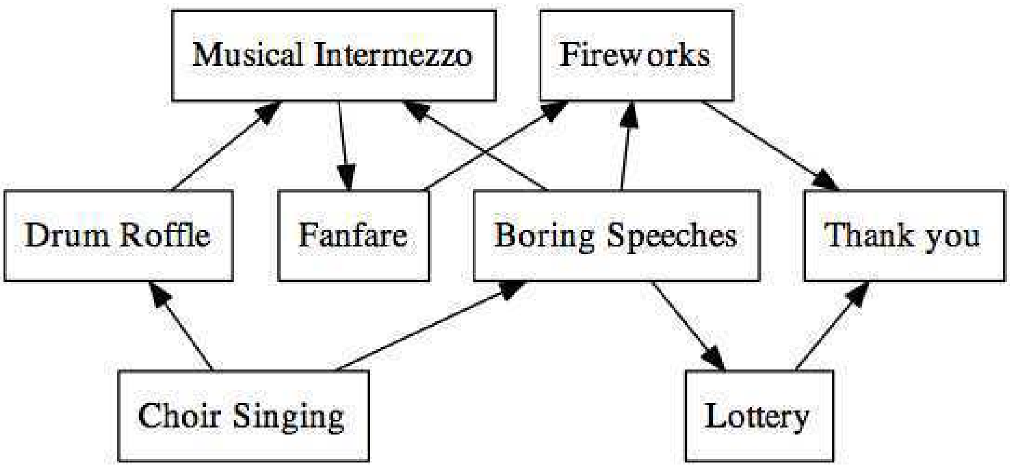
This task shows an example of a structure called a stack, which works similarly to stacking plates on top of each other. You always add new plates on top of the stack and have to remove them from the top one at a time. This type of structure is commonly referred to as a LIFO-structure – the objects that have been added last are the first to be removed. (LIFO = Last In First Out)

Ceremony

Kits:		Intermediates:	C
Castors:		Seniors:	B
Juniors:		Elite:	A

Organizing a festive day is a lot of work in Bebras City. All the events must occur in a specific order.

The diagram shows all the events that must be included. The arrows indicate that an event has to occur before another event. For example, the Musical Intermezzo can only happen after both the Drum Roffle and the Boring Speeches have finished.



Rearrange the events of the day in the box below.
Drag them into an order that follows the rules shown in the diagram.

- Fireworks
- Lottery
- Thank you
- Boring Speeches
- Drum Roffle
- Choir Singing
- Fanfare
- Musical Intermezzo

Answer:

Choir Singing
Drum Roffle
Boring Speeches
Musical Intermezzo
Fanfare
Lottery
Fireworks
Thank you

Explanation:

One way to obtain such an order is to follow the following algorithm:

While some events have not yet happened, find one such that every event pointing at it has already occurred. For example, at the beginning, “Choir Singing” is the only event that has no other event pointing at it, so it can proceed.

Then, since both “Boring Speeches” and "Drum Roffle” only had “Choir “Singing” pointing at them, and since “Choir Singing” has already happened, both of these can now take place. We can start with "Drum Roffle”, for example. And so we can continue in this manner until we have our sequence.

There are several other valid answers, depending on which event is chosen first when there are options available in the algorithm.

It’s Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL)

Solving this tasks means performing a topological sort of the graph. This means understanding the graph representation, as well as the precedence relationships.

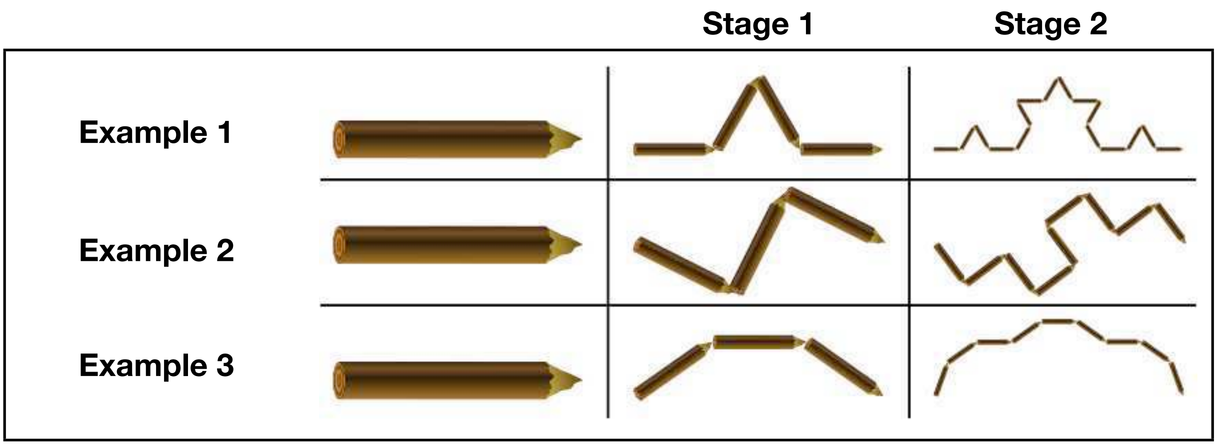
Log-art

Kits:		Intermediates:	C
Castors:		Seniors:	B
Juniors:		Elite:	A

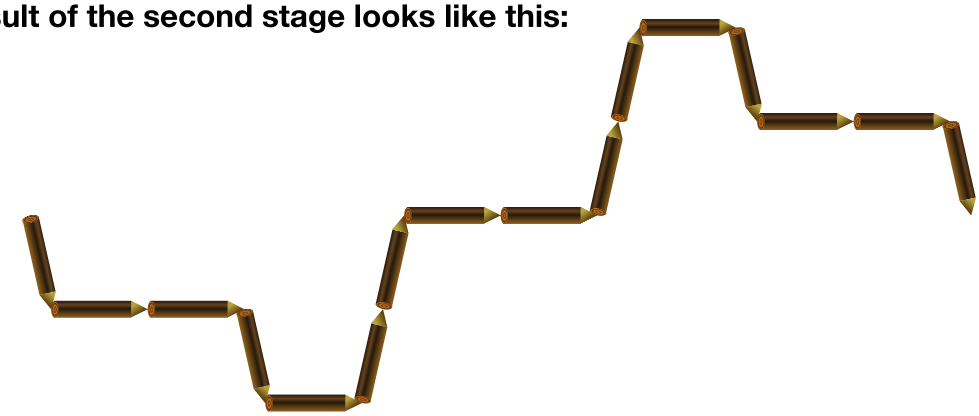
When beavers gnaw on trees they enjoy placing the pieces in a special way.

The beavers start with a single log. In stage one a big log is gnawed into smaller logs. In the next stage each individual log is again gnawed into even smaller logs but always keeping to the starting pattern. This keeps repeating.

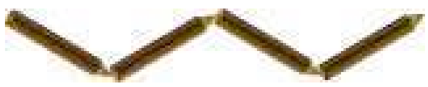
Here are three examples. On each line you see how the beaver started, the result after stage one and the result after stage two.

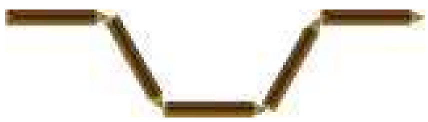


If the result of the second stage looks like this:



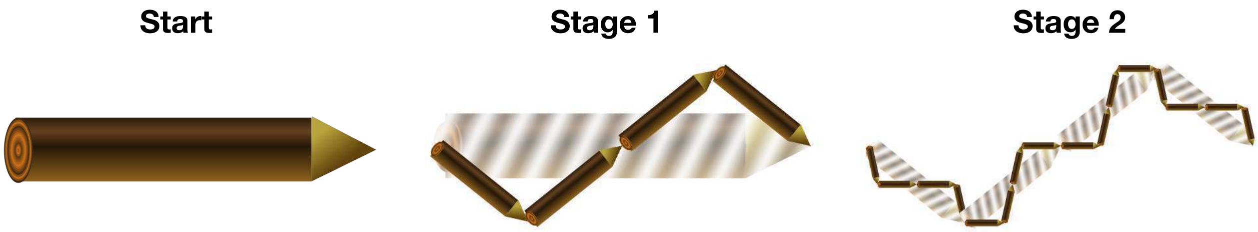
What was the first stage?





Answer:

Explanation:



It's Computational Thinking:

Concepts - Abstraction (AB), Generalisation (GE)

Computer programs represent a set of rules that can be executed on a computer. Even very simple rules can lead to complex behaviour if applied repeatedly. In this task we present the construction rules of so-called fractals. Fractals are graphics with an infinitely self-similar structured shape. Even simple rules may result in stunningly beautiful graphics.

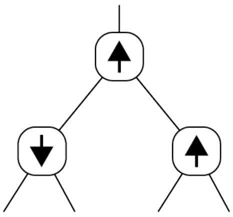
The first example shows the first 2 construction steps of Koch's snowflake curve.

Broken machines

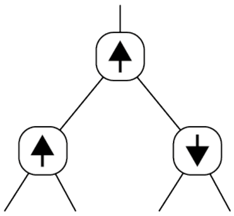
Kits:		Intermediates:	
Castors:		Seniors:	B
Juniors:		Elite:	B

Hans constructed three machines, which were all supposed to output the second largest value from a list of four numbers.

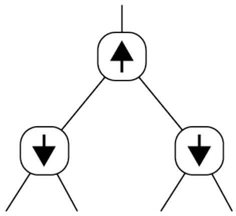
Machine 1



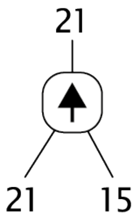
Machine 2



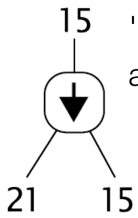
Machine 3



The machines can use two different components, called 'max' and 'min'.



'max' takes two numbers and outputs the largest



'min' takes two values and outputs the smallest

In other words, if numbers represented by a, b, c and d are input to a machine in this order, the results would be as follows:

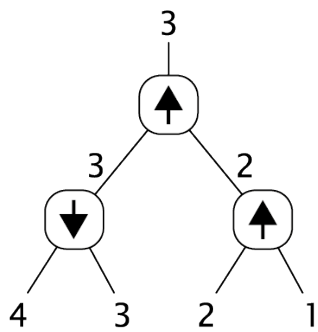
Machine 1: outputs max(min(a,b), max(c,d))

Machine 2: outputs max(max(a,b), min(c,d))

Machine 3: outputs max(min(a,b), min(c,d))

For example, if Hilda inputs the numbers 4, 3, 2, 1 into Machine 1, the output she will get is 3, which is indeed the second largest value.

However, as she continued working with the devices she quite quickly realised that none of the machines actually work. In fact, she only needed to try two number combinations in order to discover this.



Which of the following combinations did she use to prove none of the machines work?

1, 2, 4, 3 and 2, 3, 4, 1

2, 1, 3, 4 and 2, 3, 4, 1

1, 4, 2, 3 and 2, 3, 4, 1

1, 4, 2, 3 and 4, 1, 2, 3

Answer:

1, 4, 2, 3 and 2, 3, 4, 1

Explanation:

It is possible to check that for each combination of values that appears in the answers, the result will be incorrect for at least one device:

Combination	Device 1	Device 2	Device 3
1,2,4,3	X (outputs 4)		
1,4,2,3		X (outputs 4)	X (outputs 2)
2,1,3,4	X (outputs 4)		
2,3,4,1	X (outputs 4)		X (outputs 2)
4,1,2,3		X (outputs 4)	X (outputs 2)

From the table, we can see that it is enough to try the combinations 1, 4, 2, 3 and 2, 3, 4, 1 as they show that none of the devices work.

Note that we could also use other pairs of these combinations to show that the three devices fail but of the choices given there was only one solution.

Can you find a single combination for which all three devices would fail?

It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL), Evaluation (EV)

Software is critical in today's world. Computer programs are used to deliver medicine, move millions of dollars between accounts and control weather monitoring systems. A failure or a “bug” in a program can have catastrophic consequences. Therefore it is very important to test the programs carefully and systematically: we should act like prosecutors in a court-room trying to prove that the accused person is guilty, otherwise there is a high risk that our programs will not always work as expected.

As this task showed, it is not enough to only try a few values at random; you might get lucky and happen to test some exceptional data for which the system actually works. Rather, testing needs to be done in a structured and organised manner.

Websites:

http://en.wikipedia.org/wiki/Software_testing

http://listverse.com/2012/12/24/10-seriously-epic-computer-software-bugs/

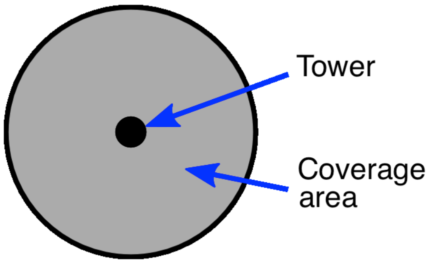
Storm proof network

Kits:
Castors:
Juniors:

Intermediates: C
Seniors: C
Elite: B

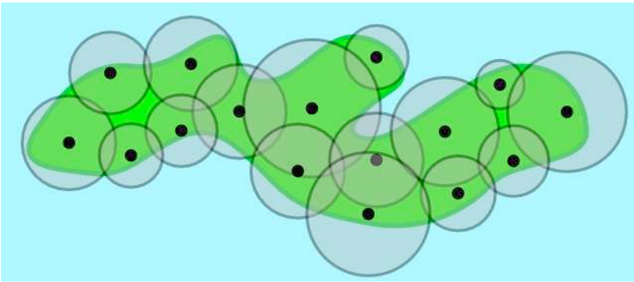
On a small green island a network of mobile phone towers is setup. Every tower covers a circular area of the island.

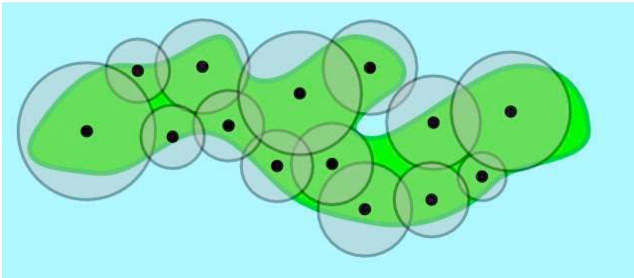
When the coverage area of two towers overlaps the towers are said to be directly connected. Towers can also be indirectly connected if there is a chain of directly connected towers between the two towers.

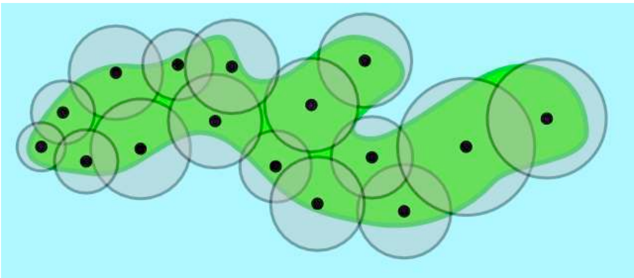


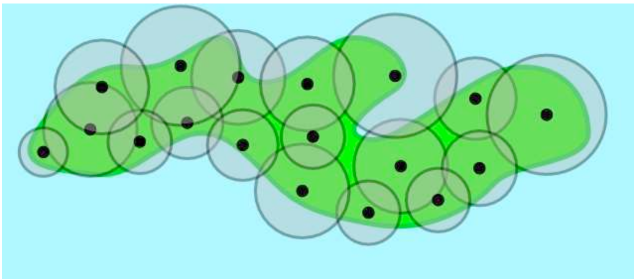
The operators want to make the network of towers Storm Proof. This means that even if one tower breaks down all other towers must still be connected, either directly or indirectly.

Which system shown below is a way to create a storm proof network on the island?

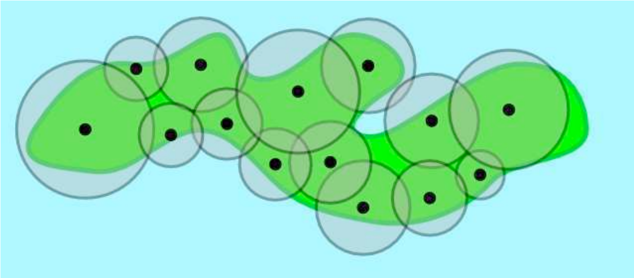








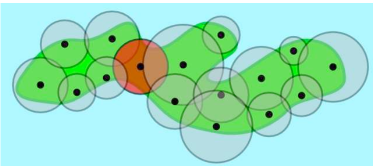
Answer:

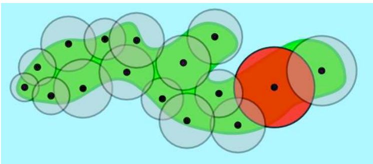


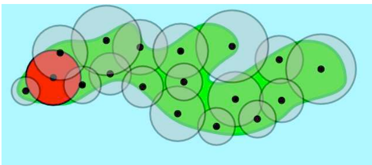
Explanation:

Following the coast, one can see that the towers are connected in a loop, if a tower breaks, a signal can still be sent between any two remaining towers.

In the other cases, there exists a tower that, if broken, will mean a signal cannot be sent between some pair of towers. One example in each case is highlighted in the pictures below. Insert images:







It's Computational Thinking:

Concepts - Abstraction (AB), Evaluation (EV), Generalisation (GE)

The placement of the towers (called vertices or nodes) and the way they are connected is called a graph or network topology. Industries need us to study these structures looking for designs that result in a system that is as reliable as possible. Similar structures can be physical or logical and they can take various shapes (e.g. ring, tree, mesh, etc.). We might then also ask other questions and evaluate properties related to different aspects of usability.

Note that, in reality the towers communicate via directed antennas and over different frequencies than mobile phones.

Footprints

Kits:		Intermediates:	
Castors:		Seniors:	C
Juniors:		Elite:	B

The instructions for a **1-tree**:

Step forward 1 step to make one footprint, go back in your own prints.

When you know how to make a 1-tree, you can learn how to make a **2-tree**:

Step forward 2 steps to make two footprints.

Turn left and make a 1-tree.

Turn right and make a 1-tree.

Go back in your own prints.

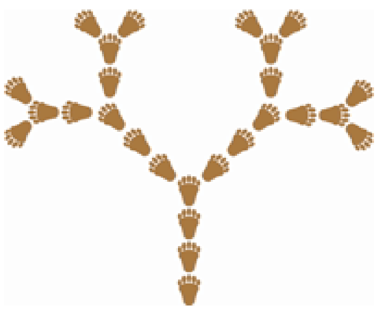
It is easy to explain how to create a **3-tree** because a 3-tree consists of 2-trees:

Step forward 3 steps to make three footprints.
Turn left and make a 2-tree.
Turn right and make a 2-tree.

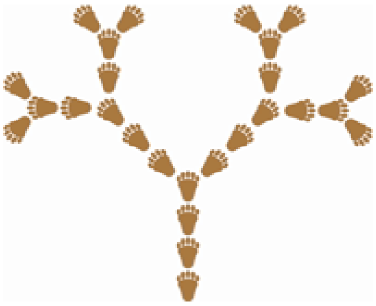
Go back in your own prints.

In a similar way you can create a 4-tree.

Which of the following trees is a proper 4-tree?



Answer:



Explanation:

This tree is a 4-tree, made of 4 steps plus two 3-trees.

It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE)

The scheme is as compact as it is clever. It works for any imaginable size of a tree.

An X-tree is printed through X steps forward plus two (X-1)-trees. These are printed through X-1 steps forward plus two (X-2)-trees …

In Computational Thinking this is called recursion where a task is explained through a simpler version of the task itself. Finally, recursion puts the task down to a base task, like printing a 1-tree: to make an X-tree, you just have to remember the way to combine (X-1)-trees into it, and only remember how to do a 1-tree.

In practice, recursively defined algorithms can be very elegant. On the other hand, overuse of recursion can produce a confusing and inefficient algorithm.

This task is also an example of a recursive fractal which can produce beautiful output.

Website:

https://en.wikipedia.org/wiki/Recursion

Puddle jumping

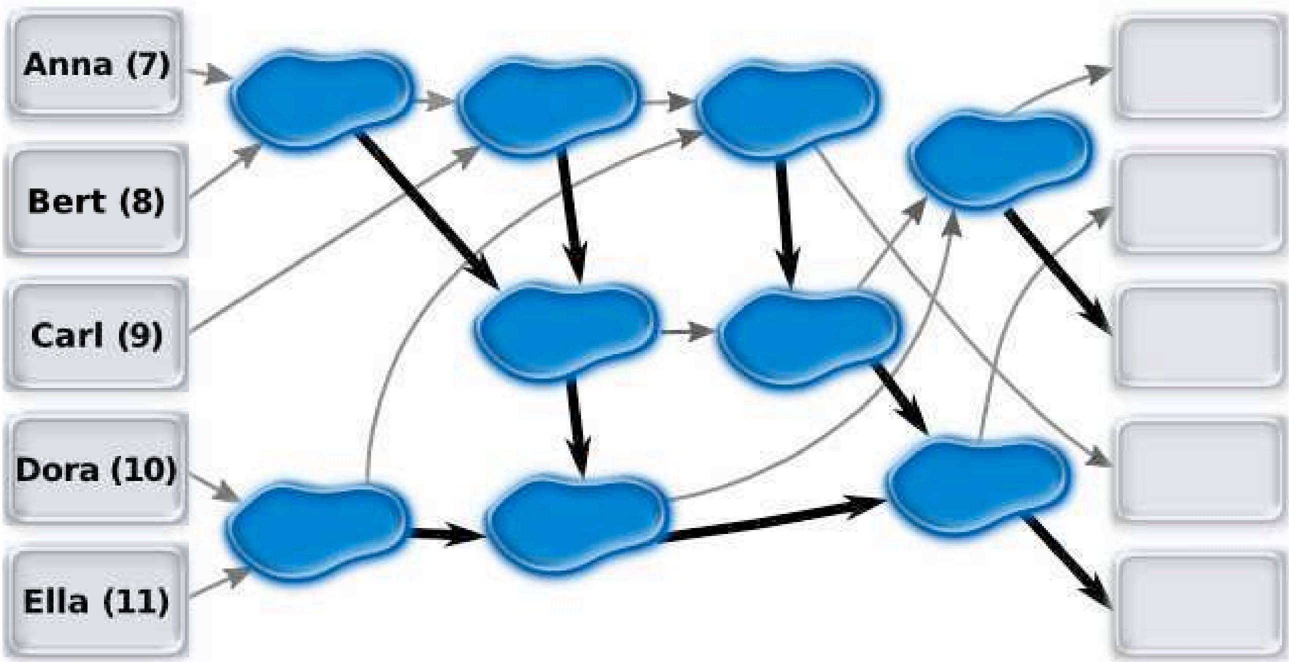
Kits:		Intermediates:	
Castors:		Seniors:	C
Juniors:		Elite:	B

Anna (age 7), Bert (age 8), Carl (age 9), Dora (age 10) and Ella (age 11) are playing a game where they jump from puddle to puddle.

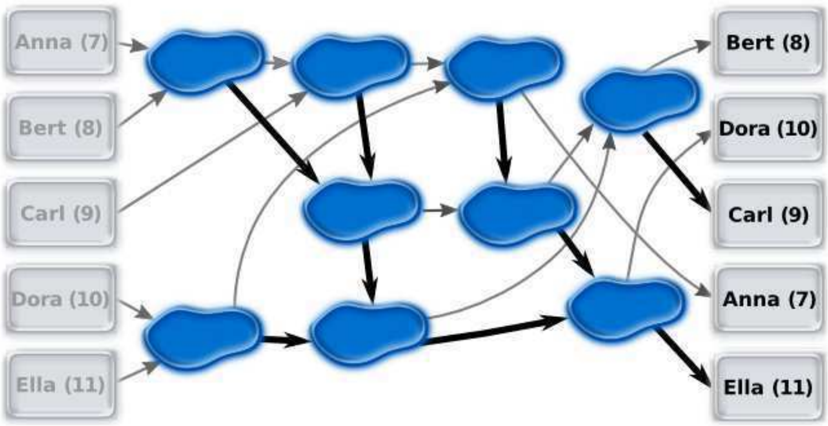
They have placed arrows between the puddles, and they all start on the left side as indicated.

When a child jumps into a puddle he or she waits for a second child to arrive. The oldest child will then jump following the thick arrow, the youngest, follows the narrow arrow.

Drag the names of the children to the right side of the field to show where each child ends.

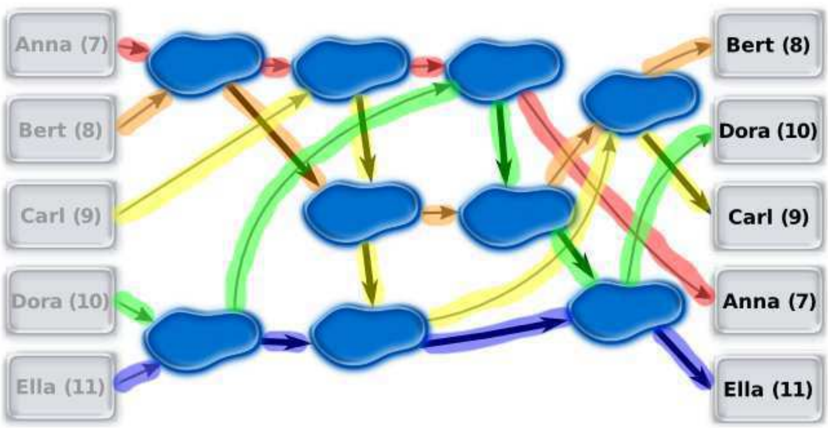


Answer:



Explanation:

By using some colourful highlighters we can trace the movement of the children and the decisions made at each puddle:



It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE)

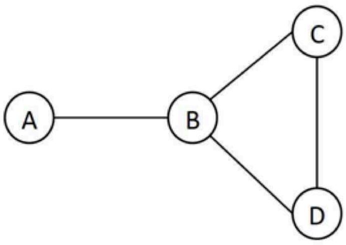
Networks consist of wires and many comparators that compare the values of the two incoming wires and then output the smaller and the larger of different inputs. If this network has comparators that are interconnected in the right way, it can sort a sequence of unsorted values. Such a network is called a sorting network. These sorting networks work in parallel and can be very efficient at sorting.

In the example task, the network of puddles does not sort because the comparators are not connected in such a way. It just takes a given input sequence and outputs a different order.

Social network

Kits:	Intermediates:
Castors:	Seniors:
Juniors:	Elite: B

Both of the pictures show the same information about friendships between beavers that live in a lodge.



	A	B	C	D
A		O		
B	O		O	O
C		O		O
D		O	O	

For example, beaver A is only friends with beaver B (and beaver B is also friends with beaver A). If beaver A wishes to become friends with beaver C, he would need to get an introduction by Beaver B. The following diagram shows the friendships between 7 beavers.

	A	B	C	D	E	F	G
A		O	O	O			
B	O		O	O			
C	O	O		O			
D	O	O	O		O		
E				O		O	O
F					O		O
G					O	O	

What is the minimum number of introductions beaver A needs in order to become friends with beaver G?

1 2 3 or 4

Answer:

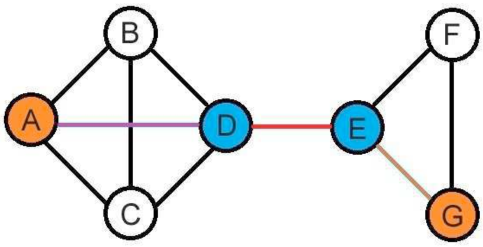
2

Explanation:

A needs two introductions at least, according to the friendship table:

	A	B	C	D	E	F	G
A		O	O	●			
B	O		O	O			
C	O	O		O			
D	O	O	O		O		
E				●		O	●
F					O		O
G					O	O	

or we can rebuild the social network as following:



It's Computational Thinking:

Concepts - Algorithmic Thinking (AL), Evaluation (EV), Generalisation (GE)

A network (or graph) is a visual notation for a human. However, an adjacency matrix is used to represent a graph in a program. With this matrix, we can further apply all the algorithms from graph theory or manipulate the nodes and links in the graph, so the transformation between graph and adjacency matrix is an important concept for computer scientists.

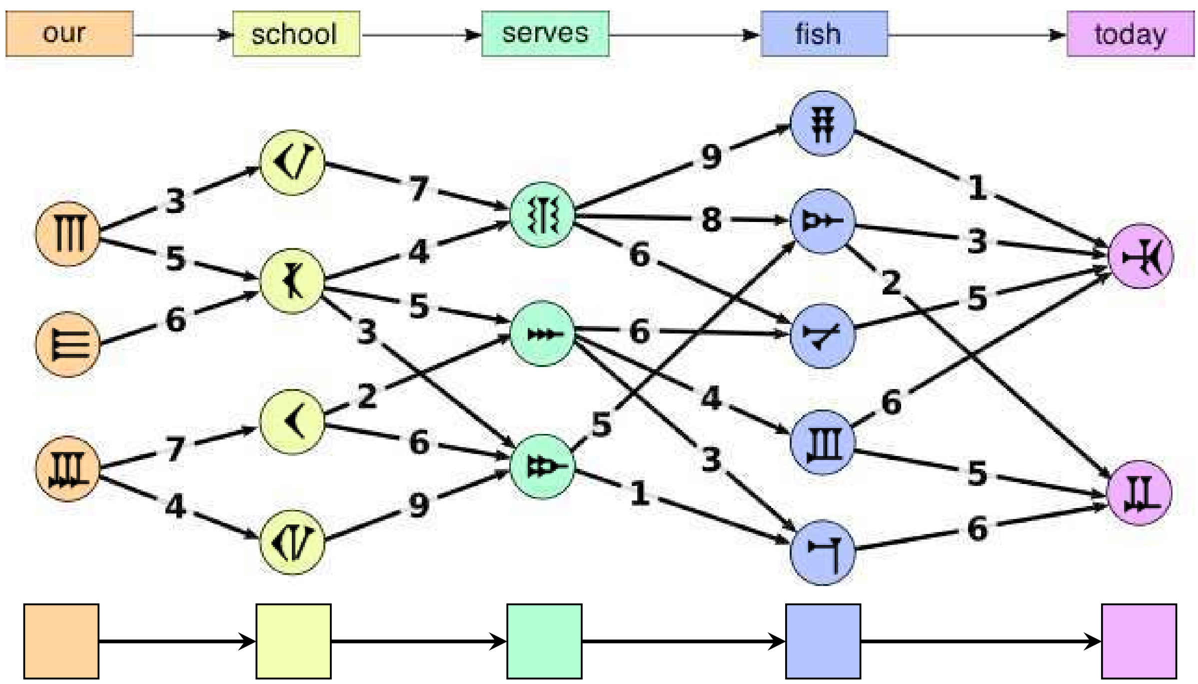
Best translation

Kits:
Castors:
Juniors:
Intermediates:
Seniors: C
Elite: C

Aeysha is trying to translate an English sentence into an ancient language. Every word translates into a foreign symbol. There are several possible symbols for each word. Betty wants to find the best translation.

Under each English word Betty has written down the possible foreign symbols. Between each pair of symbols, she has indicated how well they fit together in that order. (A higher number means a better fit.)

The best translation would be the five symbols which produce the highest total maximum score. Betty has prepared the translation of 'our school serves fish today':



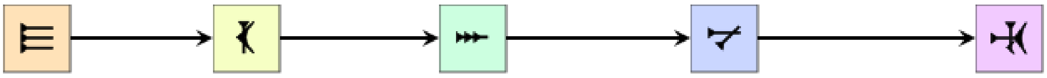
Determine the best translation.

Drag the correct symbol into the empty space underneath each word.

Answer:

The best possible score is 22.

Counting from top, choose the second, second, second, third and finally first symbol.



Explanation:

How can we find this quickly among all the valid translations? We should proceed systematically; for example word by word, from left to right. Let's say I am at the middle green point. I would like to know the best score I can get on the way to that green point. If I proceeded systematically, I have all the partial top scores to the left. So I can easily see, whether I should come from the second or the third yellow point. There is no other option! I can continue with the next green point and then go to the pale blues and proceed in the same manner. This way we can find the best score very fast. We use only 22 additions and some comparisons, instead of 36*3=108 additions, made by those who checked every possible way.

For extra fun, can you confirm that there are indeed 36 valid translations?

It's Computational Thinking:

Concepts - Abstraction (AB), Evaluation (EV), Generalisation (GE)

The smart idea to solve this quickly is called dynamic programming. It is based on a general idea of systematically building the solution from small chunks to bigger and bigger pieces. If you remember (or write down) the partial results, this can be done very quickly.

Also, this gives you a glimpse of contemporary machine translation such as used by *Google translate*. You may be surprised, but it does not depend on deep understanding of grammar. Rather, it works with enormous databases of texts in different languages, and simply put, looks for good matches. You might wonder if you can use this approach as a human to learn a new language. Unfortunately, a sufficient amount of text doesn't fit into a human brain, so you will still need to study proper grammar!

Website:

https://translate.google.com/about

Right rectangles

Kits:		Intermediates:	
Castors:		Seniors:	C
Juniors:		Elite:	C

A robot has been programmed to draw rectangles. It can execute the following instructions:

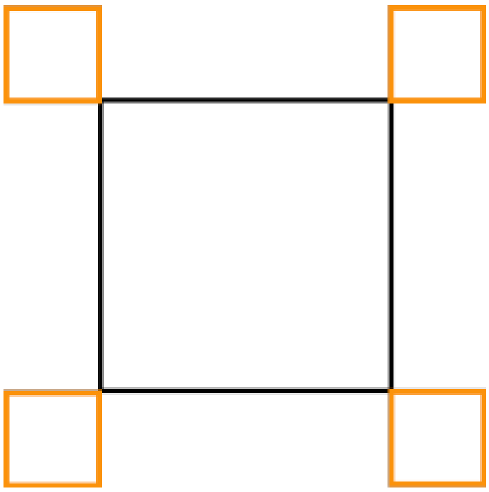
Orange	draw an orange line of length 1
Black	draw a black line of length 1
Turn	turn 90° clockwise

Besides those simple instructions the robot can also execute complex instructions by combining instructions.

If A and B are instructions (either simple or complex) the robot can do:

A, B	first execute A and then execute B
n×(B)	execute B *n*-times

The robot must draw this pattern:



Which set of instructions does NOT result in the requested drawing?

4×(2×(Orange, Turn), Orange, 3×(Black), Orange, Turn)

4×(2×(Orange, Turn), 3×(Black), 2×(Orange, Turn))

4×(3×Black, 3×(Orange, Turn), Orange)

4×(Black, 3×(Orange, Turn), Orange, 2×(Black))

Answer:

4×(2×(Orange, Turn), 3×(Black), 2×(Orange, Turn))

Explanation:

Apart from the answer, all the other sets of instructions create the desired drawing but from a different starting point.

It's Computational Thinking:

Concepts - Algorithmic Thinking (AL), Evaluation (EV)

The program of the robot is a so-called algorithm, in other words a sequence of commands. It describes how a problem (here the drawing of the figure) is solved by decomposing the problem into many small individual steps.

These individual steps are repeatedly executed when needed (here, for example, 3 x Black to draw the long black line).

If the right commands are in the correct order, we have a program that solves the problem.

Height game

Kits:	Intermediates:	
Castors:	Seniors:	
Juniors:	Elite:	C

Young beavers Amy, Beavy, Cuttree, Diggy and Eary, want to play a game with you.

They all stand in a line. Then they each count how many beavers are taller than they are both infront of them and behind. They give you the results on a slip of paper:

	No. of taller beavers	
Name	**infront**	**behind**
Amy	1	2
Beavy	3	1
Cuttree	1	0
Diggy	0	0
Eary	2	0

In what order are the beavers standing?

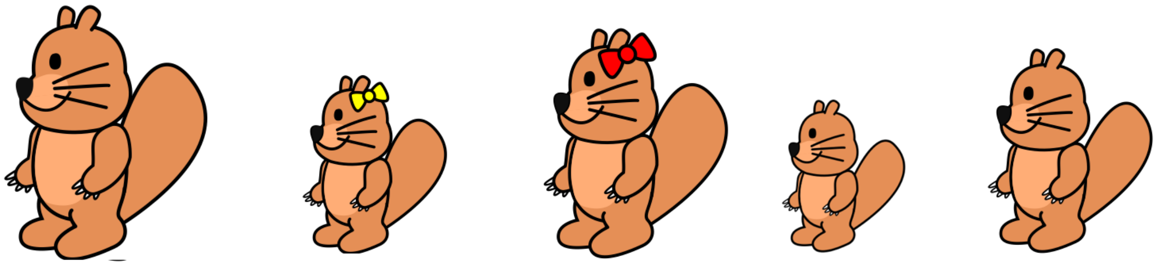
Diggy, Cuttree, Amy, Beavy, Eary

Amy, Cuttree, Diggy, Eary, Beavy

Diggy, Amy, Cuttree, Beavy, Eary

Diggy, Amy, Eary, Beavy, Cuttree

Answer:



Diggy, Amy, Cuttree, Beavy, Eary

Explanation:

First establish the height order. e.g. Diggy is tallest as there are no beavers taller than her and Cuttree is next, down to Beavy who is the shortest because there are a total of 4 beavers taller than her. It can be helpful to assign the beavers height integers like this:

Diggy(5)	Cuttree(4)	Eary(3)	Amy(2)	Beavy(1)

Now it is much easier to assign the positions: Diggy must be at the front, as all the others have beavers in front of them. Beavy must be in the fourth position because she has three beavers in front of her and one behind:

Diggy(5)			Beavy(1)	

Because Amy has two taller beavers behind her (who must be Cuttree and Eary) and Beavy is smaller than her, Amy must be in the second position:

Diggy(5)	Amy(2)		Beavy(1)	

Finally, because Eary has no taller beavers behind her, she must be behind Cuttree and thus in the last position. So we also know where Cuttree is:

Diggy(5)	Amy(2)	Cuttree(4)	Beavy(1)	Eary(3)

It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL), Evaluation (EV)

Sorting is an essential concept in computer science. The solution to many problems may require sorting as a necessary first step. It allows us to order the unordered data and simplify solving the algorithm.

Logic and computer science are deeply connected. When solving a logical problem, as well as writing a computer program, it really helps to approach the problem step by step and establish intermediate results that can then be used to solve the full problem.

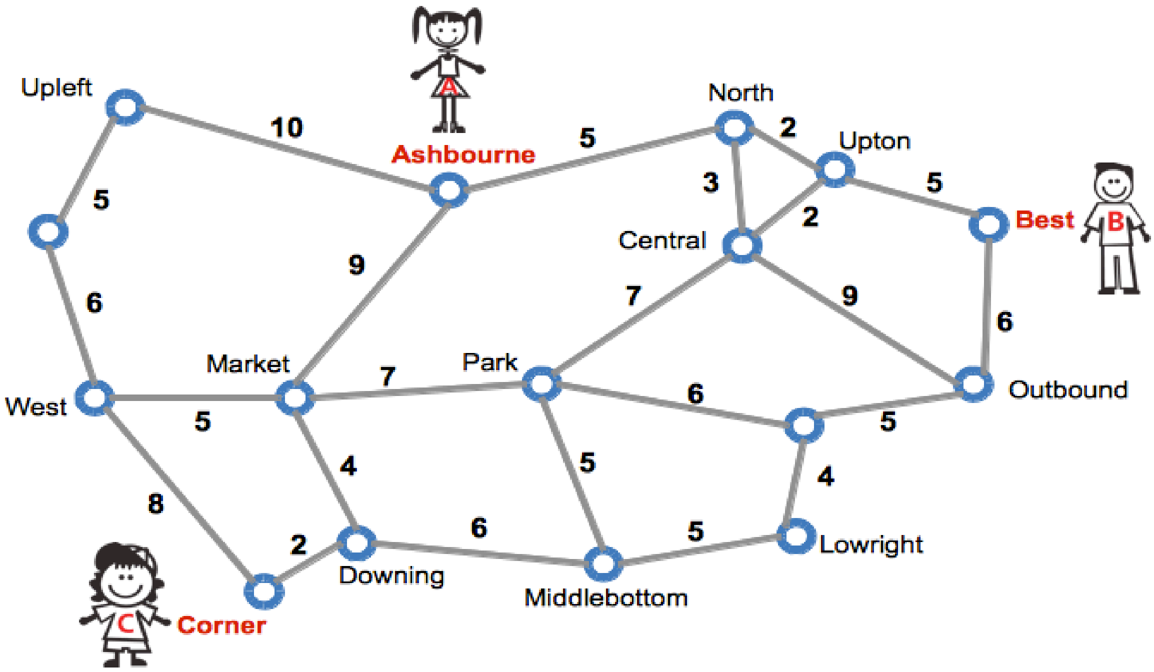
Meeting point

Kits:
Castors:
Juniors:
Intermediates:
Seniors:
Elite: C

Three friends Anne, Bernie and Clara live in a city with an underground train system.

The map of the system below shows the stations and connections between the stations.

The map also indicates how many minutes each connection takes.



Anne lives next to Ashbourne station, Bernie's nearest station is Best, and Clara's is Corner. They wish to select a station for a meeting. None of the friends should take more than 15 minutes of travel to reach the meeting point.

Which stations qualify as possible meeting points?

Click on all of the stations that would qualify as suitable meeting points.

Answer:

Ashbourne and Park are good meeting points

Explanation:

In order to solve this problem, we have to know to which stations we can travel to from each of Ashborn, Best and Corner, in less than 15 minutes. To do this, we add up the times that constitute the shortest route to any other station.

So from Ashbourne we find we can get to Ashbourne(0), (North,5), Upton(7), Best(12), Central(8), Park(15),(Market,9), Downing(13), West(14) and (Upleft,10).

We then do the same for Best and Corner, and keep the stations which are common for the three.

Alternatively we can use Dijkstra's algorithm:

Let us take Ashbourne as the starting station. We make a temporary list of stations, and their time of travel, which at the beginning is empty. First, we add (Ashbourne,0) to this list: we can travel to Ashbourne in 0 minutes. Then we take the station with the lowest minutes from the temporary list: (Ashbourne,0). Now we mark that Ashbourne takes 0 minutes. Then we add to the temporary list all the neighbours of Ashbourne, with their total times: (North,5), (Upleft,10) and (Market,9).

We then take the nearest station from the temporary list: (North,5) and declare that North is 5 minutes from Ashbourne. Then we add the neighbours of North with the sum of the times: (Upton,7) and (Central,8). We do not add Ashbourne because we already know we can get to it faster. The list is now (Upleft,10), (Market,9), (Upton,7), (Central,8). We repeat the process until all stations in the temporary list take more than 15 minutes to get to.

This gives us the minimal time for every station. After we have done the same for Best and Corner, we can keep the stations which are common for the three.

It's Computational Thinking:

Concepts - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

The mathematical concept of a relationship is heavily employed in the area of Computational Thinking. Large databases rely on so-called relational models. One way of representing relationships between elements of the same set is in a "graph": the elements are called "nodes", and "edges" are pairs of related nodes. In specialised graphs, edges may have a direction (node a is related to node b, but not vice versa) or weights. A traffic network like the one of this task can be modelled - and visualised - as a graph with weighted edges, with the weights representing the time of traveling distances. The good thing is that Computer Scientists have developed many efficient algorithms for such graphs, e.g. several ones to find shortest paths. Obviously, shortest path algorithms are useful for route planning - and many other applications.

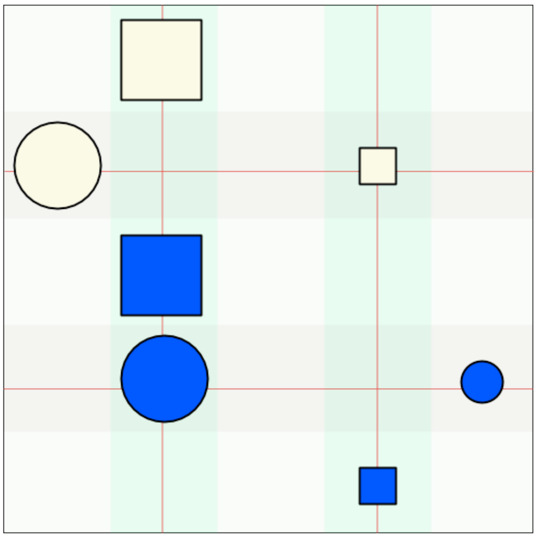
True or false

Kits:
Castors:
Juniors:
Intermediates:
Seniors:
Elite: C

Alice and Tom are playing a game of "True or False" on their colourful, magnetic whiteboard in their classroom.

Alice has stuck seven different magnetic shapes on the board.

She then makes four statements about the shape, colour, size and position of the shapes.
Only one statement is allowed to be true.
Tom must figure out which one it is.



Which of the following statements is true?

A. There are two shapes X and Y, so that X is dark blue and Y is pale yellow and X is higher than Y.
B. For all pairs of shapes X and Y, if X is a square and Y is a circle, then X is higher than Y.
C. For all pairs of shapes X and Y, if X is small and Y is big, then X is to the right of Y.
D. For all pairs of shapes X and Y, if X if light yellow and Y is dark blue, then X is below Y.

Answer:

C

Explanation:

The correct answer is C since every small shape is to the right of every large shape.
There is no dark blue shape above a light yellow shape, so A is false.
Not all squares are above every circle, so B is false.
Not all light yellow shapes are below every dark blue shape, so D is false.

It's Computational Thinking:

Concepts - Algorithmic Thinking (AL), Evaluation (EV)

The game is about determining whether logical statements are true or false. Each statement can be expressed in predicate logic. The properties of a shape X can be represented by the predicates square(X), circle(X), large(X), small(X), blue(X), and yellow(X). The relations between a pair of shapes X and Y can be represented by the predicates above(X,Y), below(X,Y) and right(X,Y). Using these predicates, the statements can be expressed formally as:

A) exist X, Y: blue(X) and yellow(Y) and above(X, Y)

B) forall X, Y: (square(X) and circle(Y)) implies above(X, Y)