

Evaluating and Adapting the ATTA for Computational Thinking Skills Assessment in Higher Education: A Pilot Study

Frances Sheridan
National College of Ireland
fsheridan@ncirl.ie

Pramod Pathak
TU Dublin
pramod.pathak@tudublin.ie

Paul Stynes
National College of Ireland
paul.stynes@ncirl.ie

Keith Quille
TU Dublin
keith.quille@tudublin.ie

Abstract

Computational Thinking (CT) Skills are growing in importance as generative AI influences the skills required for a modern workforce. A review of the literature shows a concentration of CT research at the primary and secondary education levels, with a clear research gap and limited availability of practical, validated CT assessment tools for use in Higher Education. The Algorithmic Thinking Test for Adults is a 20-item validated scale targeting several CT components with an execution time of approximately 70 minutes. A pre-pilot study using the ATTA highlighted challenges related to timing and execution time when zero responses were collected. A revised pilot study with 125 invited participants achieved a 44% response rate. Participant feedback coupled with completion rates and item non-response rates supported a need for a reduction of the 20-item scale to improve feasibility in this context. This research aims to evaluate the suitability, reliability, and relevance of the Algorithmic Thinking Test for Adults for use with undergraduate computing students and explores adaptations to improve its feasibility in time-constrained classroom settings. Iterative item removal and exploratory factor analysis produced a shortened 9-item scale with acceptable reliability indicated by Cronbach's α of .764. Findings provide preliminary evidence that a shortened ATTA with reduced execution time may improve feasibility while retaining acceptable internal consistency.

1. Introduction

As GenAI influences the skills required for a modern workforce, educators must consider the learning challenges that this presents (Romero, 2024). Researchers have now begun to consider prompt engineering a new 21st Century skill (Federiakin, Molerov, Zlatkin-Troitschanskaia, & Maur, 2024), and emphasise the importance of CT skills for AI literacy (Celik, 2023). A recent Blog post for Communications of the ACM highlights this relationship between CT skills and Prompt Engineering for computing students in Higher Education and emphasises that "These interactions between prompt engineering and computational thinking underscore the importance of computational thinking in the era of GenAI" (Erez, Mike, & Hazzan, 2024). Much of the research on CT skills to date has focused on curriculum, frameworks and interventions to support the development of CT skills (Lockwood & Mooney, 2017). However, in order to fully appreciate the impact of frameworks and interventions, we need valid and reliable assessment tools. A review of the literature shows a concentration of CT research at the primary and secondary education levels, with a clear gap in research and limited availability of practical, validated assessment tools for use in Higher Education (Haseski & Ilić, 2019; Tang, Yin, Lin, Hadad, & Zhai, 2020; Cruz Castro, Shoaib, & Douglas, 2021; Sheridan, Stynes, Pathak, & Quille, 2024). For the purpose of this research, a review of the literature was performed aiming to identify the full range of tools available to assess CT skills at all levels of education (Sheridan et al., 2024). Tools were reviewed with an emphasis on evaluating the reliability of the tools, as well as their suitability and adaptability for use in this research. Of the tools available to assess CT Skills in Higher Education, the Algorithmic Thinking Test for Adults (ATTA) (Lafuente Martínez, Lévesque, Benítez, Hardebolle, & Zufferey, 2022) stood out as the most robustly validated and reliable tool available for a Higher Education context. The ATTA is a 20-item scale validated using a series of measures including two qualitative pilot studies, expert review, and a quantitative pilot involving 289 participants where participants received a cash incentive for participation. While the scale targeted several components of CT, results found algorithmic thinking to be the

single dimension underlying the test. Questions in the ATTA are presented to participants in increasing order of difficulty. The execution time for the test is approximately 70 minutes - a factor identified by the researchers as a possible limitation.

1.1. Pre-pilot Study

A Pre-pilot study was conducted in April of the 2023/2024 academic year in two Irish Higher Education institutions, National College of Ireland (NCI) and Technological University Dublin (TUD). Year 1 Computing students were invited via email to complete the survey in their own time and no incentive was provided to students in exchange for participation. A total of 300 students were invited to participate and zero responses were received. The challenges of recruiting participants for Higher Education research are many (Khatamian Far, 2018). One challenge often encountered when recruiting and retaining Higher Education students as research participants is timing (of Ontario, Cyr, Childs, & Elgie, 2013). When considering participation in research studies, students will weigh the costs and benefits of participation. Following the survey, informal verbal feedback was gathered by lecturers regarding non-participation. The informal feedback indicated that the time of year, the duration of the survey, and the lack of incentive may have been factors that impacted participation.

1.2. Pilot Study

The response rate of the pilot study further emphasises the limitation of the execution time identified during the original ATTA validation. Research on response rates indicates that rates increase when the respondent burden (the effort required to answer the questions) is low (Denscombe, 2009). Thus, the researchers decided to narrow the scope of the pilot study to one single institution where students could be allowed to complete the survey during class time. The pilot study was carried out in Semester 1 of the following academic year, 2024/25 in NCI only. A total of 125 first-year undergraduate computing students were invited to participate. The survey, distributed electronically via email, retained the same structure as before, including demographic questions, items from the MSLQ (Pintrich & De Groot, 1988), and the full 20-item ATTA (Lafuente Martínez et al., 2022). Students were given 60 minutes during class time to complete the survey. As with the previous phase, no extrinsic incentives were offered for participation. A total of 55 responses to the survey were received. This represented 44% of the cohort of students. Of these, 11 students completed the ATTA for their own experience but opted out of the study itself and the remaining 44 responses were opted into the study. In a subsequent class, students were invited to complete a short, two-question, online feedback questionnaire. Students were asked if they had: completed the survey; started but not completed the survey; or did not even start the survey. They were also asked, in an open response question, to outline their reasons for non-completion. The feedback survey conducted in NCI received 66 responses. Of the 66, 44% stated that they completed the survey, 41% started but did not finish the survey, and the remaining 15% did not start the survey at all. Analysis of responses regarding non-completion found that 10 students cited the volume of questions, 16 said they did not have enough time, 10 said the questions were too difficult to answer, and a further 5 had difficulties understanding the questions themselves.

1.3. Research Goals

This research aims to:

1. Evaluate the suitability, reliability, and relevance of the ATTA as a CT assessment instrument for practical, time-constrained application in the context of a higher education computing classroom.
2. Identify any necessary adaptations to the ATTA to reduce the time demands placed on respondents, improving feasibility for classroom use, whilst maintaining its reliability as a measure of computational thinking in undergraduate computing students.

2. Background

Computational Thinking (CT) as a concept dates back to the work of Papert in 1980 (Papert, 2020), with a surge in interest for educational researchers following Janet Wings' seminal article in 2006 (Wing,

2006). Most recently, CT has been brought further into focus by the rapid emergence of Generative AI (GenAI) tools. According to Wing, “Computational Thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science” (Wing, 2006). However, there is still a lack of consensus on a formal definition of CT (Gouws, Bradshaw, & Wentworth, 2013; Grover & Pea, 2013). Similarly, there is some variation when considering the components of CT, but when reviewing the literature, Algorithmic Thinking, Decomposition, Abstraction, Pattern Recognition and Evaluation are commonly featured as core components of CT (Lafuente Martínez et al., 2022).

2.1. Computational Thinking Assessment

As the area of CT skills development has grown in popularity, a number of reviews have attempted to synthesise the existing literature on CT assessment tools. A review by de Araujo et al. in 2016 (Araujo, Santos, Andrade, Guerrero, & Dagienė, 2017) highlighted inconsistencies in CT assessment owing to the lack of a generally accepted CT definition and noted difficulties with replication of research as a result of insufficient detail in study details. Another review by Haseski and Ilic (Haseski & Ilić, 2019) identified a lack of emphasis on affective variables such as self-efficacy and called for more research on CT assessment tools for Higher Education. Other reviews called out a lack of evidence for the validity and reliability of CT assessment tools, and the difficulties in applying K-12 based CT frameworks to Higher Education (Tang et al., 2020; Cruz Castro et al., 2021).

2.2. Tool Type and Focus

Tools for measuring CT Skills take several forms, ranging from tests (Anistyasari, Ekohariadi, Buditjahjanto, & Hidayati, 2021; Salac, Thomas, Butler, & Franklin, 2021; Zhang, Wong, & Pan, 2021; Brackmann et al., 2017; Guggemos, Seufert, & Román-González, 2019; Román-González, Pérez-González, & Jiménez-Fernández, 2017), self-reporting surveys (Sondakh, Osman, & Zainudin, 2019) and rubrics (Von Wangenheim et al., 2018; Clarke-Midura, Silvis, Shumway, Lee, & Kozlowski, 2021; Leela, Chookeaw, & Nilsook, 2019; Metcalf et al., 2021; Werner, Denner, Campe, & Kawamoto, 2012) among others. Of these, tests are the most popular form at all levels of education largely due to their ease of automation and scalability for larger studies making them easily replicable in new contexts. Some tools focus on specific programming constructs (Basu, Rutstein, Xu, & Shear, 2020; Chang, Sun, Wu, & Guizani, 2018; Parker et al., 2022; Zapata-Cáceres, Martín-Barroso, & Román-González, 2020; DEMİR-KAYMAK, DUMAN, RANDLER, & HORZUM, 2022; Diago, González-Calero, & Yáñez, 2022; Fanchamps, Slangen, Specht, & Hennissen, 2021; Hu, Chen, Yuadi, & Chen, 2022), while others centre around CT specific concepts (Bati, Yetişir, Çalışkan, Güneş, & Gül Saçan, 2018; Lee & Cho, 2021; Lafuente Martínez et al., 2022; Kalliopi & Michail, 2019; Koh, Nickerson, Basawapatna, & Repenning, 2014; El-Hamamsy et al., 2022) such as algorithmic thinking, problem decomposition and abstraction. While programming constructs are the most common focus of tools used at primary and secondary level, in Higher Education the focus shifts to CT specific concepts.

2.3. Tool Reliability

The evidence in support of reliability and validity of CT assessment tools varies greatly. Some publications provide statistical data regarding the reliability of the tool used such as internal consistency using Cronbachs α (Anistyasari et al., 2021; Çetin, Otu, & Oktaç, 2020; Çoban & Korkmaz, 2021; Fanchamps et al., 2021; Finke et al., 2022; Zhang et al., 2021), or coefficient α (Miller et al., 2013; Peteranetz, Morrow, & Soh, 2020), or Cohens κ for inter-rater reliability (Boom, Bower, Siemon, & Arguel, 2022; Kalliopi & Michail, 2019; Metcalf et al., 2021; Rowe et al., 2021; Yin, Hadad, Tang, & Lin, 2020) for construct development. Others provided evidence of the validity of the tool using methods such as comparison with another tool or measure (Araujo et al., 2017; BUBICA & BOLJAT, 2022; Masiulionytė-Dagienė & Jevsikova, 2022; Çoban & Korkmaz, 2021), comparison with academic achievement (DEMİR-KAYMAK et al., 2022; Mukasheva & Omirzakova, 2021; Román-González, Pérez-González, Moreno-León, & Robles, 2018), reporting on pilot studies (Ambrósio, Xavier, & Georges, 2014; Clarke-Midura et al., 2021; Lee & Cho, 2021; Leifheit et al., 2019; Palts & Pedaste, 2019; Parker

et al., 2022), or expert reviews (Lafuente Martínez et al., 2022; Lai, 2021; Li, Xu, & Liu, 2021; Sondakh et al., 2019; Tsai, Chien, Wen-Yu Lee, Hsu, & Liang, 2022; Zapata-Cáceres et al., 2020).

2.4. Summary

There remains a significant limitation in CT assessment research: many tools lack public accessibility and detailed documentation, restricting research replicability. Taking account of this, along with tool type, focus, and evidence of reliability and validity, three potential candidates for use in this research emerged: the Computational Thinking Test (CTt) (Román-González et al., 2017), the Algorithmic Thinking Test for Adults (ATTA) (Lafuente Martínez et al., 2022), and the Holistic Assessment of Computational Thinking (HiACT) (Sondakh et al., 2019). CTt is widely validated but primarily designed for K-12, while HiACT, aimed at higher education, emphasizes self-reported approaches rather than performance. ATTA, explicitly developed for adults and validated through psychometric analysis with 289 incentivised participants, assesses both knowledge and skills. Given these attributes, ATTA was identified as a potentially suitable tool for use in this study.

3. Method

The goal of this research is to pilot the ATTA within an undergraduate Higher Education Computing context in order to evaluate its suitability, reliability, and relevance for application within this context. In doing so, this research aims to identify any necessary adaptations to the ATTA with a view to reducing the time demands placed on students while maintaining its effectiveness as a measure of CT skills. Pre-pilot and pilot studies were conducted using the ATTA in an undergraduate Higher Education Computing context in 2023/2024 and 2024/2025 academic years (Sheridan, Stynes, Pathak, & Quille, 2025). The pre-pilot received a response rate of zero and feedback was used to inform the design of the pilot study the following academic year.

3.1. Data Collection & Analysis

During the pilot study, responses were collected using Microsoft Forms and the collected responses were then logged using Microsoft Excel. For the purpose of reporting, the original numbering of items in the ATTA was retained. This numbering system retained the original numbers used during the development of the ATTA. As questions appear in the survey in ascending order of difficulty, the numbers do not appear in sequence. The 20 items in the survey are numbered 27, 25, 26, 20, 13, 23, 7, 16, 6, 8, 21, 9, 17, 2, 11, 12, 24, 1, 19 and 3. The full set of questions and answers for the ATTA can be found in Appendix A. Those who opted not to participate in the research study were removed from the data set. Participants had also consented to making their module results available for comparison. The results of their first programming assignment were added to the data set at this point. Each record was anonymised, assigning a unique id number to each participant. The responses to each question of the ATTA were reviewed and correct answers were marked as 1 and incorrect as 0. Items which had been skipped were marked as 0 and also coded as non-response at this point. The total score for each participant was calculated with each question assigned the same value. Descriptive Statistics were calculated for the full data set and Pearson's correlation coefficient was calculated to examine the relationship between the ATTA and Programming Project scores. To confirm the internal consistency of the ATTA, Cronbach's α was calculated.

3.2. Item Non-Responses

Items with high non-response rates were examined. While high non-response can indicate problematic items, decisions regarding item removal were not based on missingness alone. The ATTA is designed such that the questions appear in ascending order of difficulty. Although the authors do not explicitly provide a psychometric justification for this, their item development process emphasised a need for test accessibility for both novices and experts and deliberately sought to include a range of difficulties across items. As the intention of this analysis is to reduce the scale for use in a time constrained classroom setting, the level of missingness was carefully considered as it might indicate an item either being unreachable or too difficult for the target participants. Items toward the end of the survey with low or zero variance coupled with high levels of non-response were removed at this point in order to

facilitate further analysis.

3.3. Item Reduction

Kaiser-Mayer-Olkin (KMO) and Bartlett's Test of Sphericity (BTS) were then calculated to evaluate the suitability of the data for factor analysis. Initial results indicated sufficient correlation between items for factor analysis as evidenced by a significant BTS. However, the overall KMO value was unacceptable. This can sometimes be an indication that the ratio of responses to variables is unsuitable for factor analysis. The ATTA contains 20 items and a total of 37 responses were received. The literature on the ratio of sample size to number of variables varies, from 5:1 to 30:1 (Cohen, Manion, & Morrison, 2002).

An iterative item reduction process was then undertaken with the objective of increasing the KMO to a level acceptable for factor analysis. During each iteration, the performance of the item was evaluated using multiple criteria, including non-response patterns, sampling adequacy, inter-item correlations, examining the KMO, BTS, and Cronbach's α before and after the removal of each item to evaluate the impact of item removal. Exploratory factor analysis was conducted using Principal Component Analysis using Direct Oblimin rotation with Kaiser normalisation.

Firstly, items toward the end of the scale that had little to no variance and high non-response rates were considered unreached items and thus eliminated. KMO and BTS statistics were then re-examined alongside the anti-image matrix and individual Measures of Sampling Adequacy (MSA). An item with MSA below .5 is deemed unsuitable for factor analysis as it has very weak relationships with other variables and is poorly predicted by other items (Kaiser, 1974). Subsequently, items with MSA below .5 were eliminated in turn starting with the item with the lowest MSA. KMO and BTS were re-examined after each elimination. Next, inter-item correlations were examined iteratively, removing items that showed weak correlations with other items or predominantly negative correlations (DeVellis & Thorpe, 2021). Items with most correlations below .3 or with mostly negative correlations were removed in turn, re-examining KMO and BTS after each removal.

3.4. Factor Analysis

Once the KMO and BTS were at acceptable levels, exploratory factor analysis was conducted using Principal Component Analysis with Direct Oblimin rotation and Kaiser normalisation. Examination of Eigenvalues and Scree Plot indicated a 2-factor solution. Items that did not support this model or that mapped uniquely to a third factor were examined further and removed if appropriate.

4. Results

Through an iterative item reduction process, 11 of the 20 items were removed from the scale. Table 1 identifies each of the removed items, the reason for its removal, and the Cronbach's α and KMO values after its removal.

4.1. Data Collection and Analysis

There were 44 participants in total. Of those, 7 subsequently withdrew from their programme of study making assessment results unavailable. Thus, the remaining 37 responses were considered in the analysis alongside assessment results. Overall scores were calculated for each participant. The mean score was 30.54 as a percentage of correct answers and standard deviation was 16.95. Pearson's correlation Coefficient of the overall ATTA score and assessment result was .59.

4.2. Item Non-Response

The final 2 items in the scale, items 19 and 3 had high non-response rates of 59% and 51% respectively. These items were considered unreachable and removed from the scale. Once these items were removed, Cronbach's α on the remaining 18 items was calculated as .722. KMO on this 18-item scale was .465 which is unacceptable for factor analysis.

4.3. Item Reduction

MSA for each of the individual items was calculated using an anti-image correlation. Item 17 was excluded at this point as it had a low MSA value of .196 and a high non-response rate of 32%. After the removal of item 17, KMO increased to .556 which is considered poor but not unacceptable. Examination of individual item MSA for the remaining items led to the iterative removal of item 2, item 26, and item 7 as each of these had MSA values below .5. This brought KMO to an acceptable value of .691 and retained only items with an individual MSA above .5. Next, the correlation matrix was examined. Items 1 and 13 which had few or no correlation with other items above .3 were both removed. The KMO and Cronbach's α for the remaining 12 item scale were calculated as .697 and point .727 respectively. Item 11, which had only negative correlations with the remaining items was removed next. After removal of item 11, KMO increased to an acceptable .706 and Cronbach's α increased to .782.

4.4. Factor Analysis

With KMO at .706, a level considered acceptable, exploratory factor analysis was conducted and the component matrix was generated using Principal Component Analysis where 4 factors were extracted with eigenvalues greater than 1. The scree plot was also generated. Inspection of the Scree plot showed a clear bend after the second component supporting the case for a 2-factor solution rather than the 4-factor solution indicated by the component matrix. Further examination of the component matrix showed that the third and fourth components were supported by only one item each: the third component by item 8; and the fourth component by item 21. When both items were removed, the final value for Cronbach's α was calculated at .764 and KMO at .71.

Table 1 – Sequential Item Removal Summary

Item Removed	Reason for Removal	α	KMO
19	Unreachable: Non response rate: 59%	.720	.465
3	Unreachable: Non response rate: 51%	.722	.465
17	MSA Below .5 (.196) & High Non-response rate (32%)	.724	.556
2	MSA below .5 (.243)	.733	.605
26	MSA below .5 (.401)	.735	.628
7	MSA below .5 (.486)	.713	.670
1	All correlations below .3	.715	.670
13	Only 1 correlation above .3	.727	.697
11	Only Negative Correlations	.782	.706
8	Factor Analysis: only item mapped to 3rd component	.766	.708
21	Factor Analysis: only item mapped to 4th component	.764	.710

The final solution presents two factors with 6 items loaded on the first component and 2 on the second. According to the component correlation matrix, correlation between the two factors is .283 which indicates that the factors are only weakly related with low to moderate correlation. This implies that while the two factors are not completely independent, they are also not measuring exactly the same thing. The original validation of the ATTA aimed to address several components of CT but found that algorithmic thinking was the single dimension underlying the test. Examination of the categories of CT targeted by each item during the original development of the ATTA shows that Algorithmic Thinking is either the first or second most endorsed category of all of the 6 items loaded on the first component. This aligns with the finding of the original validation that algorithmic thinking was the main underlying component. For the two items that load on the second component, Algorithmic Thinking was the most endorsed category for both items also. However, the second most endorsed category for those items was Evaluation. This overlap of categories may provide some explanation for the low correlation between the two factors.

4.5. Summary

When considering the full 20-item scale across 37 participant responses, Pearson's correlation Coefficient of the overall ATTA score and programming assessment result was .59. Calculating the score for each participant on the revised 9-item ATTA, Pearson's correlation coefficient of revised participant scores and programming assessment results was calculated as .56. Both Pearson's values of .59 and .56 indicate a moderate to strong correlation between the ATTA result and programming assessment result. The similarity between the Pearson's value for both the 20-item and 9-item scales implies that the removal of the 11 items has had little impact on the criterion validity of the scale for this sample.

The original validation of the 20-item ATTA scale calculated Cronbach's α of .854 (Lafuente Martínez et al., 2022, p. 1457) indicating very good reliability well above the threshold of .7 (Cronbach, 1951). The revised 9-item scale resulting from this study has a Cronbach's α value of .764, below the .854 of the full 20-item scale but still above the .7 threshold. This suggests that although the reduced scale demonstrates a slight decrease in internal consistency when compared to the original 20-item scale, it still maintains acceptable reliability.

Preliminary assessment of the execution time for the revised 9-item ATTA places execution time between 30 and 40 minutes. This is approximately half of the time required to complete the original 20-item ATTA. This, combined with improved internal consistency, and comparable correlation coefficients, indicates that the 9-item ATTA may offer a more efficient alternative with limited evidence of reduced criterion validity for this sample.

5. Conclusion & Future Work

This research evaluated the suitability, reliability, and feasibility of the ATTA CT assessment instrument for practical, time-constrained application in the context of a higher education computing classroom. While the ATTA had previously been robustly validated as a tool for measuring Computational Thinking skills, findings of the pre-pilot and pilot phases of this research confirm the execution time as a significant practical limitation. Low participation and student feedback in both the pre-pilot and pilot along with high levels of item non-response in the pilot indicate that the full 20-item scale is difficult to implement in a time-constrained setting.

A shortened 9-item ATTA was produced through an iterative process of item reduction based on non-response patterns, sampling adequacy, inter-item correlations and factor analysis. This 9-item scale appears to retain acceptable internal consistency with Cronbach's α of .764 and satisfactory sampling adequacy indicated by KMO of .71. These results suggest that a shortened ATTA may provide a more feasible and practical tool for classroom use without substantially comprising reliability or validity.

However, there are several limitations which must be acknowledged. This study was conducted with a relatively small sample size ($n=37$) from a single institution which limits the generalisability of the findings. Further validation of the reduced scale is required with a larger sample size across multiple institutions before it can be recommended for wider adoption.

The next phase of this research is to administer the reduced 9-item ATTA across multiple institutions with a larger sample size to further evaluate its reliability, validity and generalisability. Further psychometric validation such as confirmatory factor analysis is required to strengthen the validity of the revised scale. Future studies will also explore how student ATTA performance is associated with other relevant constructs such as programming self-efficacy. Overall, this work contributes to the long-term goal of developing a practical, evidence-based CT Skills assessment tool for use in CT research in Higher Education.

6. References

- Ambrósio, A. P., Xavier, C., & Georges, F. (2014). Digital ink for cognitive assessment of computational thinking. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–7). Spain: IEEE.
- Anistyasari, Y., Ekohariadi, E., Buditjahjanto, I. P. A., & Hidayati, S. C. (2021). Exploring the psy-

- chometric properties of computational thinking assessment in introductory programming. In *2021 international e-engineering education services conference (e-engineering)* (pp. 88–93). Jordan: IEEE.
- Araujo, A. L. S. O., Santos, J. S., Andrade, W. L., Guerrero, D. D. S., & Dagiené, V. (2017). Exploring computational thinking assessment in introductory programming courses. In *2017 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). Indianapolis, USA: IEEE.
- Basu, S., Rutstein, D., Xu, Y., & Shear, L. (2020). A principled approach to designing a computational thinking practices assessment for early grades. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 912–918). Portland, OR, USA: ACM.
- Bati, K., Yetişir, M. I., Çalışkan, I., Güneş, G., & Gül Saçan, E. (2018). Teaching the concept of time: A steam-based program on computational thinking in science education. *Cogent Education*, 5(1), 1507306.
- Boom, K.-D., Bower, M., Siemon, J., & Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. *Education and Information Technologies*, 27(6), 8289–8310.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th workshop on primary and secondary computing education* (p. 65–72). The Netherlands: ACM.
- BUBICA, N., & BOLJAT, I. (2022). Assessment of computational thinking—a croatian evidence-centered design model. *Informatics in Education*, 21(3), 425–463.
- Celik, I. (2023). Exploring the determinants of artificial intelligence (ai) literacy: Digital divide, computational thinking, cognitive absorption. *Telematics and Informatics*, 83, 102026.
- Çetin, İ., Otu, T., & Oktaç, A. (2020). Adaption of the computational thinking test into turkish. *Turkish Journal of Computer and Mathematics Education*, 11(2), 343–360.
- Chang, Z., Sun, Y., Wu, T.-Y., & Guizani, M. (2018). Scratch analysis tool (sat): a modern scratch project analysis tool based on antlr to assess computational thinking skills. In *2018 14th international wireless communications & mobile computing conference (iwcmc)* (pp. 950–955). Cyprus: IEEE.
- Clarke-Midura, J., Silvis, D., Shumway, J. F., Lee, V. R., & Kozlowski, J. S. (2021). Developing a kindergarten computational thinking assessment using evidence-centered design: the case of algorithmic thinking. *Computer Science Education*, 31(2), 117–140.
- Çoban, E., & Korkmaz, Ö. (2021). An alternative approach for measuring computational thinking: Performance-based platform. *Thinking Skills and Creativity*, 42, 100929.
- Cohen, L., Manion, L., & Morrison, K. (2002). *Research methods in education*. routledge.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3), 297–334.
- Cruz Castro, L., Shoaib, H., & Douglas, K. (2021). Computational thinking frameworks used in computational thinking assessment in higher education. a systematized literature review. In *Asee annual conference*.
- DEMİR-KAYMAK, Z., DUMAN, İ., RANDLER, C., & HORZUM, M. B. (2022). The effect of gender, grade, time and chronotype on computational thinking: Longitudinal study. *Informatics in Education*, 21(3), 465–478.
- Denscombe, M. (2009). Item non-response rates: a comparison of online and paper questionnaires. *International journal of social research methodology*, 12(4), 281–291.
- DeVellis, R. F., & Thorpe, C. T. (2021). *Scale development: Theory and applications*. Sage publications.
- Diago, P. D., González-Calero, J. A., & Yáñez, D. F. (2022). Exploring the development of mental rotation and computational skills in elementary students through educational robotics. *International Journal of Child-Computer Interaction*, 32, 100388.
- El-Hamamsy, L., Zapata-Cáceres, M., Barroso, E. M., Mondada, F., Zufferey, J. D., & Bruno, B. (2022). The competent computational thinking test: Development and validation of an unplugged com-

- putational thinking test for upper primary school. *Journal of Educational Computing Research*, 60(7), 07356331221081753.
- Erez, Y., Mike, K., & Hazzan, O. (2024, aug 29). Leveraging computational thinking in the era of generative ai. *BLOG@CACM*, Communications of the ACM. Retrieved from <https://cacm.acm.org/blogcacm/leveraging-computational-thinking-in-the-era-of-generative-ai/> (Online blog post)
- Fanchamps, N. L., Slangen, L., Specht, M., & Hennissen, P. (2021). The impact of sra-programming on computational thinking in a visual oriented programming environment. *Education and Information Technologies*, 26(5), 6479–6498.
- Federiakina, D., Molero, D., Zlatkin-Troitschanskaia, O., & Maur, A. (2024). Prompt engineering as a new 21st century skill. In *Frontiers in education* (Vol. 9, p. 1366434).
- Finke, S., Kemény, F., Sommer, M., Krnjic, V., Arendasy, M., Slany, W., & Landerl, K. (2022). Unravelling the numerical and spatial underpinnings of computational thinking: a pre-registered replication study. *Computer Science Education*, 32(3), 1–22.
- Gouws, L., Bradshaw, K., & Wentworth, P. (2013). First year student performance in a test for computational thinking. In *Proceedings of the south african institute for computer scientists and information technologists conference* (pp. 271–277). East London, South Africa: Association for Computing MachineryNew YorkNYUnited States.
- Grover, S., & Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1), 38–43.
- Guggemos, J., Seufert, S., & Román-González, M. (2019). Measuring computational thinking—adapting a performance test and a self-assessment instrument for german-speaking countries. In *International association for development of the information society* (pp. 183–191). Italy: ERIC.
- Haseski, H. I., & Iliç, U. (2019). An investigation of the data collection instruments developed to measure computational thinking. *Informatics in Education*, 18(2), 297–319.
- Hu, C.-C., Chen, M.-H., Yuadi, I., & Chen, N.-S. (2022). The effects of constructing robot-based storytelling system on college students’ computational thinking skill and technology comprehension. In *2022 24th international conference on advanced communication technology (icact)* (pp. 496–500). Korea: IEEE.
- Kaiser, H. F. (1974). An index of factorial simplicity. *psychometrika*, 39(1), 31–36.
- Kalliopi, K., & Michail, K. (2019). Assessing computational thinking skills at first stages of schooling. In *Proceedings of the 2019 3rd international conference on education and e-learning* (pp. 135–139). Spain: Association of Computing Machinery.
- Khatamian Far, P. (2018). Challenges of recruitment and retention of university students as research participants: Lessons learned from a pilot study. *Journal of the Australian Library and Information Association*, 67(3), 278–292.
- Koh, K. H., Nickerson, H., Basawapatna, A., & Repenning, A. (2014). Early validation of computational thinking pattern analysis. In *Proceedings of the 2014 conference on innovation & technology in computer science education* (pp. 213–218).
- Lafuente Martínez, M., Lévêque, O., Benítez, I., Hardebolle, C., & Zufferey, J. D. (2022). Assessing computational thinking: Development and validation of the algorithmic thinking test for adults. *Journal of Educational Computing Research*, 07356331211057819.
- Lai, R. P. (2021). Beyond programming: A computer-based assessment of computational thinking competency. *ACM Transactions on Computing Education (TOCE)*, 22(2), 1–27.
- Lee, K., & Cho, J. (2021). Computational thinking evaluation tool development for early childhood software education. *JOIV: International Journal on Informatics Visualization*, 5(3), 313–319.
- Leela, S., Chookeaw, S., & Nilsook, P. (2019). An effective microlearning approach using living book to promote vocational students’ computational thinking. In *Proceedings of the 2019 the 3rd international conference on digital technology in education* (pp. 25–29).
- Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., & Ninaus, M. (2019). Development of a questionnaire on self-concept, motivational beliefs, and attitude towards pro-

- programming. In *Proceedings of the 14th workshop in primary and secondary computing education* (pp. 1–9).
- Li, Y., Xu, S., & Liu, J. (2021). Development and validation of computational thinking assessment of chinese elementary school students. *Journal of Pacific Rim Psychology*, 15, 18344909211010240.
- Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? a systematic literary review. *arXiv preprint arXiv:1703.07659*.
- Masiulionytė-Dagienė, V., & Jevsikova, T. (2022). Assessing computational thinking: The relation of different assessment instruments and learning tools. In *International conference on informatics in schools: Situation, evolution, and perspectives* (pp. 66–77).
- Metcalf, S. J., Reilly, J. M., Jeon, S., Wang, A., Pyers, A., Brennan, K., & Dede, C. (2021). Assessing computational thinking through the lenses of functionality and computational fluency. *Computer Science Education*, 31(2), 199–223.
- Miller, L. D., Soh, L.-K., Chiriacescu, V., Ingraham, E., Shell, D. F., Ramsay, S., & Hazley, M. P. (2013). Improving learning of computational thinking using creative thinking exercises in cs-1 computer science courses. In *2013 ieee frontiers in education conference (fie)* (pp. 1426–1432).
- Mukasheva, M., & Omirzakova, A. (2021). Computational thinking assessment at primary school in the context of learning programming. *World Journal on Educational Technology: Current Issues*, 13(3), 336–353.
- of Ontario, H. E. Q. C., Cyr, D., Childs, R., & Elgie, S. (2013). *Recruiting students for research in postsecondary education: A guide*. Higher Education Quality Council of Ontario Toronto.
- Palts, T., & Pedaste, M. (2019). Tasks for assessing computational thinking skills at secondary school level. In *International conference on innovative technologies and learning* (pp. 216–226).
- Papert, S. A. (2020). *Mindstorms: Children, computers, and powerful ideas*. Basic books.
- Parker, M. C., Garcia, L., Kao, Y. S., Franklin, D., Krause, S., & Warschauer, M. (2022). A pair of aces: An analysis of isomorphic questions on an elementary computing assessment. In *Proceedings of the 2022 acm conference on international computing education research-volume 1* (pp. 2–14).
- Peteranetz, M. S., Morrow, P. M., & Soh, L.-K. (2020). Development and validation of the computational thinking concepts and skills test. In *Proceedings of the 51st acm technical symposium on computer science education* (pp. 926–932).
- Pintrich, P. R., & De Groot, E. V. (1988). Motivated strategies for learning questionnaire. *Journal of Educational Psychology*.
- Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in human behavior*, 72, 678–691.
- Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction*, 18, 47–58.
- Romero, M. (2024). Lifelong learning challenges in the era of artificial intelligence: A computational thinking perspective. *arXiv preprint arXiv:2405.19837*.
- Rowe, E., Almeda, M. V., Asbell-Clarke, J., Scruggs, R., Baker, R., Bardar, E., & Gasca, S. (2021). Assessing implicit computational thinking in zoombinis puzzle gameplay. *Computers in Human Behavior*, 120, 106707.
- Salac, J., Thomas, C., Butler, C., & Franklin, D. (2021). Investigating the role of cognitive abilities in computational thinking for young learners. In *Proceedings of the 17th acm conference on international computing education research* (pp. 2–17).
- Sheridan, F., Styne, P., Pathak, P., & Quille, K. (2024). Computational thinking assessment tools: A systematic review of the literature (initial findings). In *Proceedings of the 2024 conference on united kingdom & ireland computing education research* (pp. 1–1).
- Sheridan, F., Styne, P., Pathak, P., & Quille, K. (2025). Work in progress: Computational thinking skills assessment in higher education. In *2025 ieee engineering education world conference (edunine)* (pp. 1–4).

- Sondakh, D. E., Osman, K., & Zainudin, S. (2019). Holistic assessment of computational thinking for undergraduate: Reliability and convergent validity. In *Proceedings of the 2019 11th international conference on education technology and computers* (pp. 241–245).
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.
- Tsai, M.-J., Chien, F. P., Wen-Yu Lee, S., Hsu, C.-Y., & Liang, J.-C. (2022). Development and validation of the computational thinking test for elementary school students (ctt-es): Correlate ct competency with ct disposition. *Journal of Educational Computing Research*, 07356331211051043.
- Von Wangenheim, C. G., Hauck, J. C., Demetrio, M. F., Pelle, R., da Cruz Alves, N., Barbosa, H., & Azevedo, L. F. (2018). Codemaster–automatic assessment and grading of app inventor and snap! programs. *Informatics in Education*, 17(1), 117–150.
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd acm technical symposium on computer science education* (pp. 215–220).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Yin, Y., Hadad, R., Tang, X., & Lin, Q. (2020). Improving and assessing computational thinking in maker activities: The integration with physics and engineering learning. *Journal of Science Education and Technology*, 29(2), 189–214.
- Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In *2020 ieee global engineering education conference (educon)* (pp. 1905–1914).
- Zhang, S., Wong, G. K., & Pan, G. (2021). Computational thinking test for lower primary students: Design principles, content validation, and pilot testing. In *2021 ieee international conference on engineering, technology & education (tale)* (pp. 345–352).

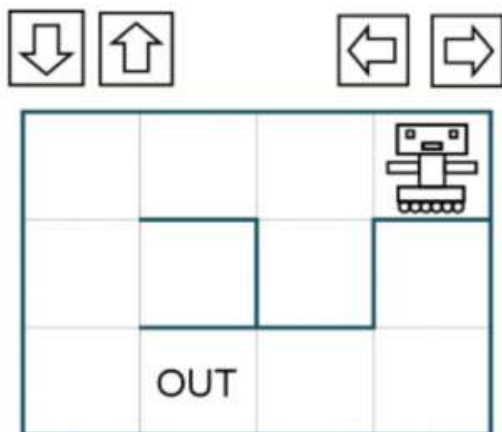
Appendix A: Survey Instrument

ATTA – Algorithmic Thinking Test for Adults

INSTRUCTIONS: Below you will find some questions that attempt to evaluate your algorithmic thinking skills. Each question is a different problem that you have to solve. Simply answer each question as best as you can. Some questions require you to select a choice, while others ask you to write a short text. Some questions may look easy and others may look difficult to you, that's fine. If you think a certain question is very difficult, don't worry, you can make a guess and try the next one. You can use a set of paper and a pen if you want. You don't need to code or use any kind of software. Thanks for your participation!

#27

This robot is programmed through a series of arrow commands ("*left*", "*right*", "*up*", "*down*"). Each arrow command causes the robot to either move one step (if the move in that direction is possible) or not to move (if the move in that direction is not possible).



Imagine the robot has a series of four commands, which it will repeat over and over again.

1	2	3	4
---	---	---	---

You can put any arrow that you want ("*left*", "*right*", "*up*", "*down*") in each command, and you may repeat the same arrow in the series of commands if you want. What should be those 4 commands in order for the robot to reach the square called "OUT" and get out of the maze?

1 _____ 2 _____ 3 _____ 4 _____

SOLUTIONS:

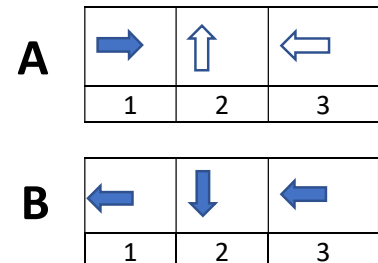
Left, left, down, right

/ right, left, left, down

/ down, right, left, left

/ right, left, left, down

#25



You have this initial set of three arrows labelled A. And, by applying some functions, you want to get this final set of arrows labelled B.

The first time you apply a function, this is applied to the three arrows; but, the following functions are applied to n-1 arrows. That is, the second function is applied only to the first two arrows; and the third function is applied only to the first arrow.

See the functions that one can apply to obtain the final set of arrows:

Function 1: turn the arrows 45 degrees anti-clockwise

Function 2: turn the arrows 180 degrees

Function 3: switch colors (turn white arrows into blue arrows, and blue arrows into white arrows)

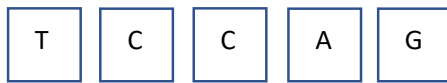
What is the sequence of functions that applied to the initial set of arrows will get you the final set of arrows?

- A) function 3; function 3; function 2
- B) function 2; function 3; function 3
- C) function 3; function 2; function 1
- D) function 3; function 2; function 3

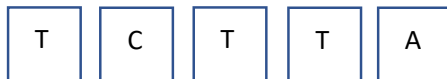
SOLUTION: D

#26

You have the following DNA sequence:



And you want to get the following sequence:



You can implement the following operations:

Swap (): it will swap one character for another. For example, *swap (A,G)* will turn the sequence AAGT into GGAT

Insert (): it will insert a specified character at the beginning of the sequence. For example, *insert (A)* will turn the sequence GT into AGT

Delete (): it will delete all specified characters. For example, *delete (A)* will turn the sequence AAGT into GT

What operations will turn the initial DNA sequence into the one we want?

- A) Delete (G); then Insert (T); then Swap (C,T)
- B) Swap (C,T); then Insert (T); then Delete (G)
- C) Insert (T); then Swap (C,T); then Delete (G)
- D) Insert (T); then Delete (G); then Swap (C,T)

SOLUTION:

B

#20

After finishing your long paper, you just discovered a mistake you made. All 4's should be 7's and all 7's should be 4's. To amend this, you can use an editor that will replace any sequence of characters with another sequence.

In order to describe the replacement you would make with the editor, please fill in the blanks:

Replace all _____ with _____; then replace all _____ with _____; then replace all _____ with _____

SOLUTION:

Replace all __4's__ with __XXs (any variable)__; then replace all __7's__ with __4's__; then replace all __XXs(variable mentioned before)__ with __7's__ (4's and 7's positions can be reversed here)

#13 A collection of elements is shown below with 2 dimensions: rows and columns. A black square indicates a 1, and a white square indicates 0. Imagine that we run two operations on the collection of elements. First, we replace column 4 by column 2; and then we replace row 2 by row 4. What will be the result?

		Column			
		1	2	3	4
Row	1				
	2				
	3				
	4				

A.

		Column			
		1	2	3	4
Row	1				
	2				
	3				
	4				

B.

		Column			
		1	2	3	4
Row	1				
	2				
	3				
	4				

C.

		Column			
		1	2	3	4
Row	1				
	2				
	3				
	4				

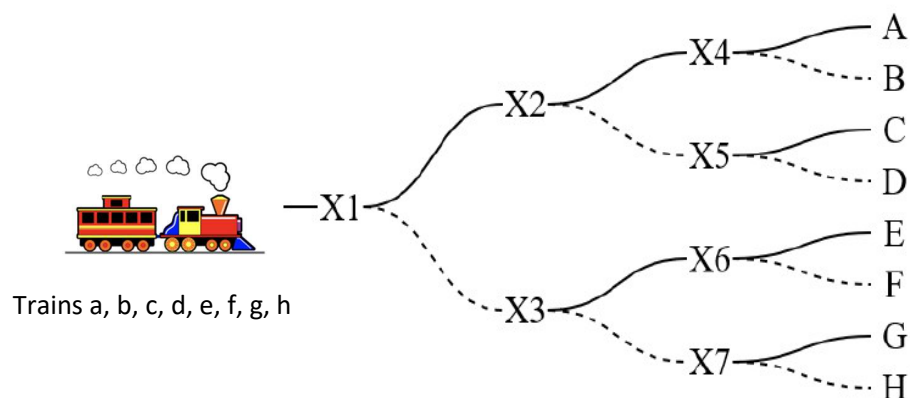
D.

		Column			
		1	2	3	4
Row	1				
	2				
	3				
	4				

SOLUTION: B

#23

Eight trains (named a, b, c, d, e, f, g, h) enter the circuit from left to right. In this circuit there are 7 switches (X1 to X7) with two positions: up or down. Each switch is initially set to direct trains up. After a train passes a switch, the switch changes from up to down, or from down to up. Train A needs to go to station A, train b to station B, train c to station C, and so on.



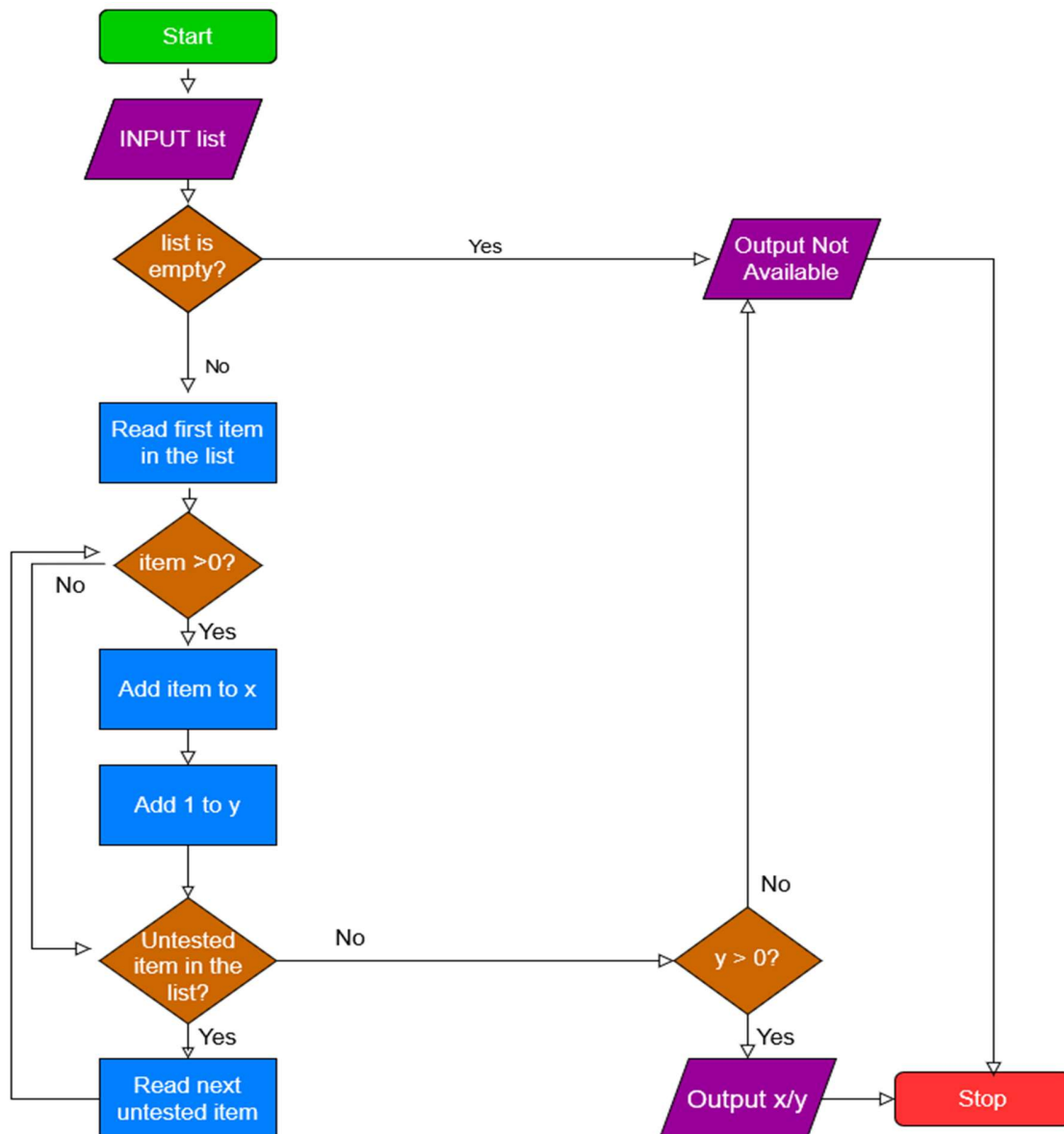
The railroad director needs to ensure that all the trains go to their correct stations, so what is the correct order for the trains to pass through switch X1?

____, ____, ____, ____, ____, ____, ____, ____

SOLUTION: A, E, C, G, B, F, D, H

#7

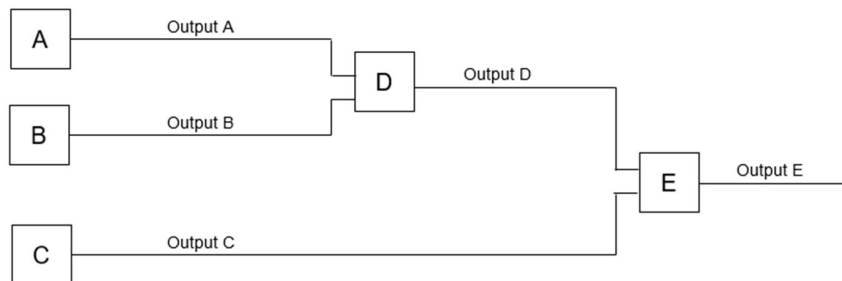
Olivia is studying the climate in her village and, over the last year, she has measured the temperature at the same time every day. She has created a list with all the temperatures she has collected so far. She prepares the following algorithm to analyze her list. What is the output she will get?



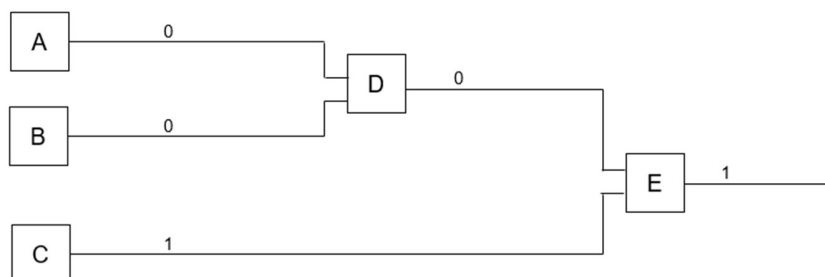
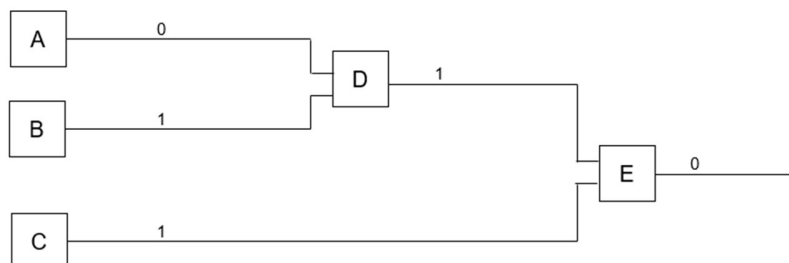
- A. The percentage of positive temperatures in the list
- B. The average of the positive temperatures in the list
- C. The number of positive temperatures in the list
- D. The sum of the positive temperatures in the list

SOLUTION: B

#16 The following scheme represents a circuit of nodes. A, B, C, D and E are nodes of this circuit. Nodes fire and pass their signal on to the next node on their right (e.g., node A gives its output to node D, which gives its output to node E). Nodes' output can be either 1 or 0.



We see the circuit produces the following states:



We know that Node D and E ALWAYS follow the same algorithm. See the possible algorithms:

Algorithm 1	If inputs are (1,1)	Outputs 1	Else: outputs 0
Algorithm 2	If inputs are (1,0) or (0,1)	Outputs 1	Else: outputs 0
Algorithm 3	If inputs are (1,1)	Outputs 0	Else: outputs 1

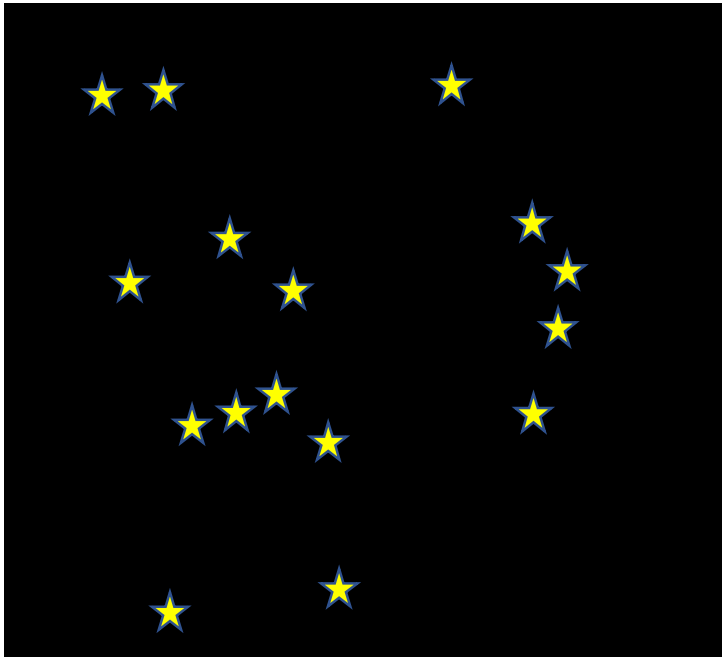
Which algorithm node D and E are both following?

- A) Algorithm 1
- B) Algorithm 2
- C) Algorithm 3
- D) This problem has no solution

SOLUTION: B

6

You are carrying out a series of astronomical observations on the Orion constellation. You want to observe, using the same telescope, all the stars included in this constellation:



Setting up a new position for the telescope takes time which you want to optimize; so, you decide that you will observe all the stars following the shortest path that connects all of them. To do that, imagine someone proposes the following algorithm:

1. Start from a random star
2. If there's still a star to observe, go to the nearest non-observed star

What is fundamentally wrong with this algorithm?

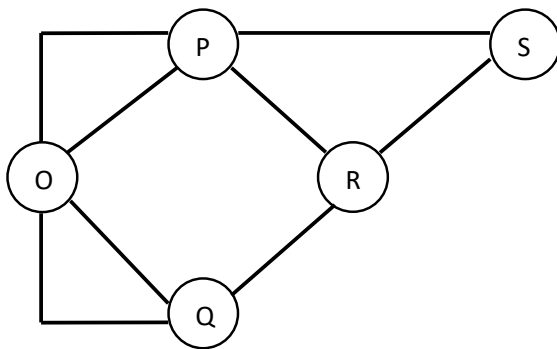
- A. The algorithm may eventually choose longer paths
- B. The starting star should not be randomly chosen, but should be a one on the center of the map
- C. The algorithm will leave non-observed stars in the path
- D. There's nothing wrong with this algorithm, it will always choose the shortest path

SOLUTION: A

8

This is a path composed of 5 vertices: O, P, Q, R, S . Every vertex is connected to other vertices through one or more lines.

You want to connect all the vertices of this path with your pencil in such a way that you pass only once through ALL THE LINES of this sketch and without lifting your pencil. You can pass more than once through every vertex, though. Where would you start the path?



- A. Always on a vertex with the lowest number of lines
- B. Always on a vertex with an even number of lines
- C. Always on a vertex with an odd number of lines
- D. One cannot connect all the vertices in figures like this

SOLUTION: C

21

We have two methods to represent a group of people. In both methods, for each person a circle is drawn. In method 1, we align people horizontally if they are connected on WhatsApp, and we align them vertically if they are connected on Instagram. In figure 1, two people are connected on WhatsApp (aligned horizontally); and three are connected on Instagram (aligned vertically).

In method 2, we draw a line between two people if they are connected either on WhatsApp or Instagram. According to method 2, the same group as above would be represented as shown in Figure 2.

A new group of six people is shown in Figure 3 according to method 1.

Which of the four diagrams **A**, **B**, **C** or **D** would represent this group of people according to method 2?

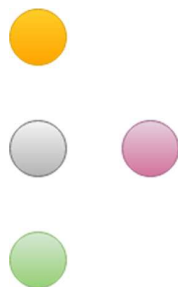


Figure 1

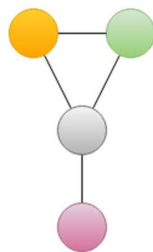


Figure 2

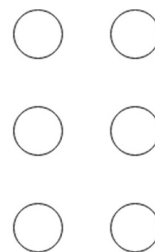
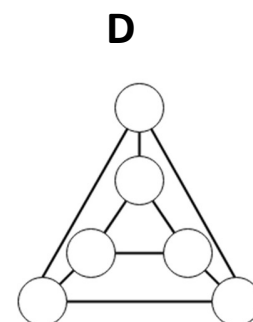
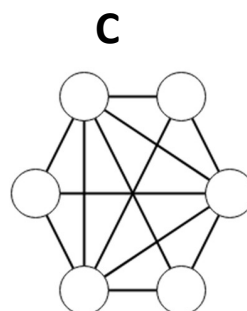
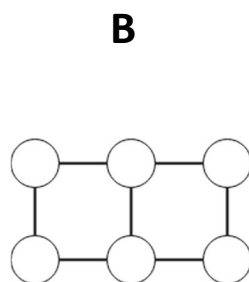
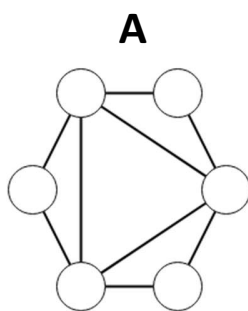


Figure 3



SOLUTION: D

9

We have this game:

There are four players (P, Q, R, S): one of them is pointing at him/herself, and three of them are pointing at any other player.

Player P gets the ball.

While the ball-holder isn't pointing at him/herself:

The ball-holder passes the ball to the player that s/he is pointing at.

If the ball-holder is pointing at him/herself, s/he drops the ball and the game finishes.

We can state that:

- A. This game will go on forever
- B. This game will eventually stop because someone will drop the ball
- C. This game will stop when player P gets the ball again
- D. We don't know whether this game will ever stop

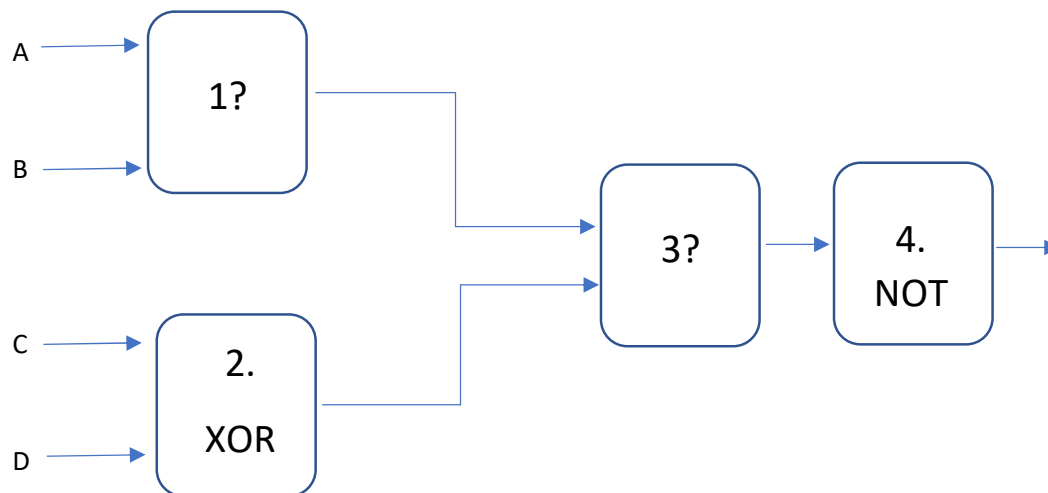
SOLUTION: D

#17

An “OR” gate outputs 1 if it receives one or more inputs with value 1. Else, it outputs 0.

A “XOR” gate outputs 1 if it has two inputs with different values (0, 1) or (1, 0). Else, it outputs 0.

A “NOT” gate inverts its input (it outputs 1 if the input is 0, and vice versa).



The inputs A, B, C and D can either be 0 or 1. We want this circuit to check that $A=B$ and that $C=D$. That is, the final output of this circuit will be 1 if $A=B$ and if $C=D$; otherwise, the output will be 0. We already know that gate 2 is a XOR and gate 4 is a NOT. What gates should we put in positions 1 and 3 of this circuit?

1_____

3_____

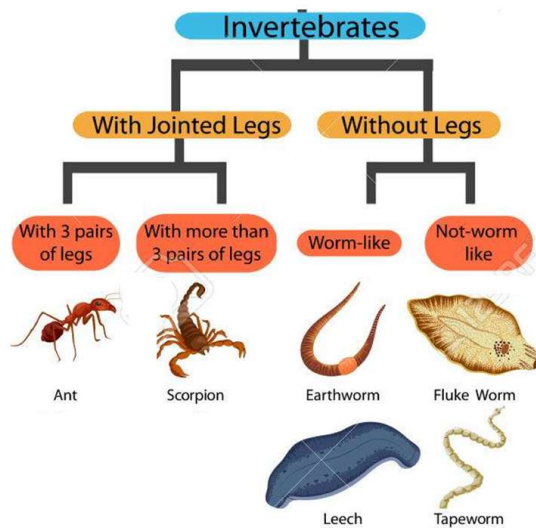
SOLUTION:

1: XOR

3: OR

2

Imagine that someone picks one of these animals and you have to guess it posing only binary questions. Binary questions are those whose answer can only be yes or no, like “does it have jointed legs?” Assuming that you follow an optimal strategy that reduces the number of questions, what is the average number of questions necessary to guess an animal uniformly picked at random from this chart?

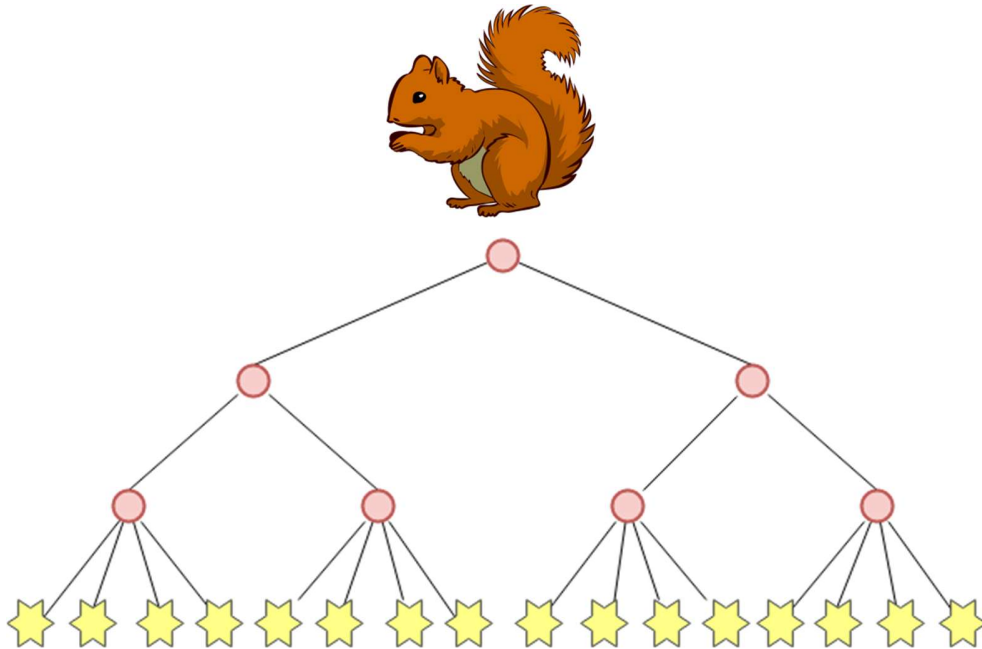


- A. 2.33
- B. 2.67
- C. 3
- D. 6

SOLUTION: B

11

Let's examine the minimum distance the squirrel must travel to pick up all the stars. How many branches does it need to go through until it picks up the last star?

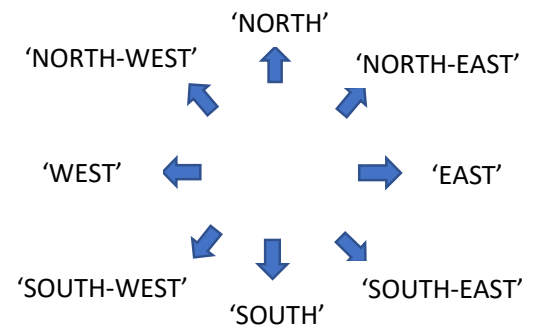
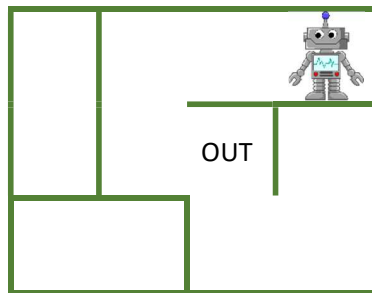
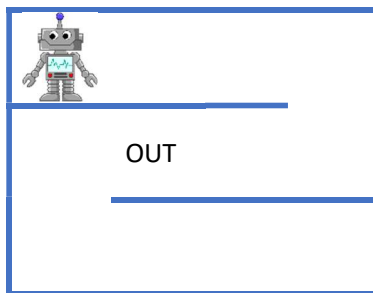


- A. Number of stars x levels of the tree
- B. 2 x number of branches – number of levels of the tree
- C. (Number of levels of the tree)²
- D. Number of branches + number of levels of the tree

SOLUTION: B

12

Describe one single algorithm that will get both robots out of their mazes in exactly 3 instructions. Both robots execute first instruction 1, then instruction 2, and then instruction 3.



Fill in the blanks with one the following eight words, where these coordinates are set according to how you see the maze:

1. If the 'out' square is to the SOUTH-EAST of the robot, keep advancing squares to the EAST until the robot collides with a wall (it cannot go in that direction any further)

Else go to the _____ until the robot collides with a wall

2. When the robot collides

If the 'out' square is to the SOUTH-WEST of the robot, keep advancing to the _____ until the robot collides with a wall

Else go to the SOUTH until the robot collides with a wall

3. When the robot collides

If the 'out' square is to the _____ of the robot, keep advancing to the WEST until the robot finds the out square

Else go to the _____ until the robot finds the out square

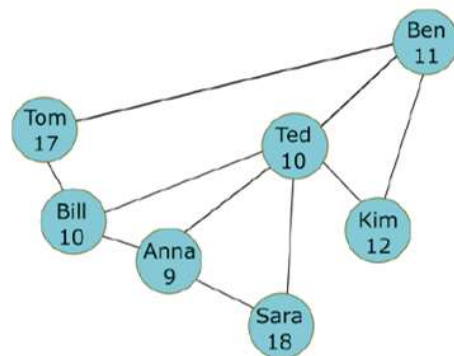
SOLUTION: WEST, SOUTH, WEST, EAST

24

This diagram shows the relationship between 7 students in a book-sharing club. Their names and ages are shown in the image. Students who are friends are connected through a line.

The club has some regulations for members:

When you receive a book, you read it (if you haven't already done so) and then pass it to the youngest friend who has not read it yet. If, however, all your friends have read the book then you should pass it to the friend who first gave it to you.



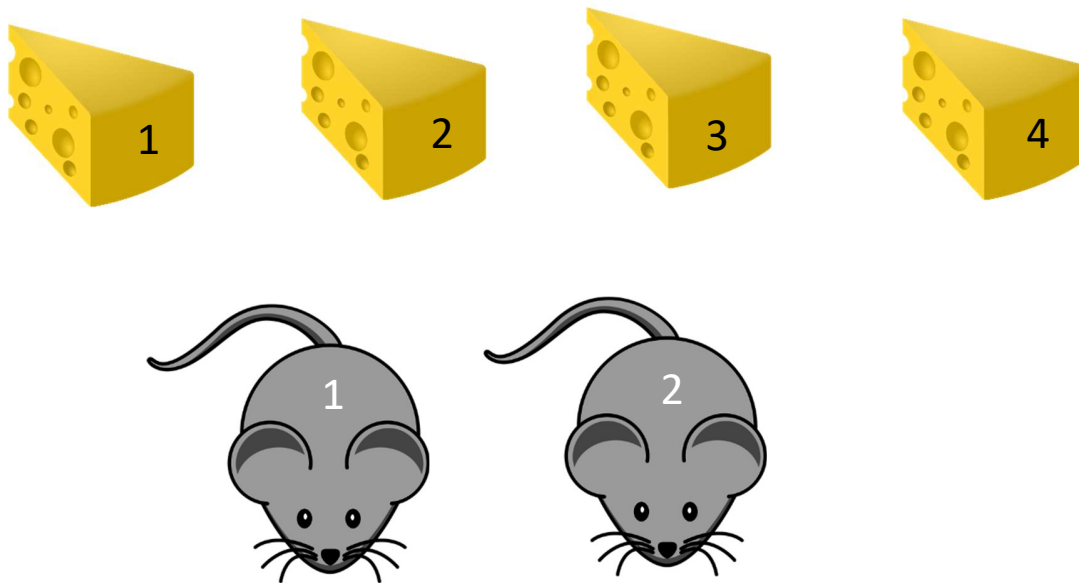
Ben has read a new book and wants to share it with his friends. Who will read the book last?

SOLUTION:

KIM

1

You have four different cheeses and one of them has a tasteless and odorless bacterium. This cheese causes symptoms of food-poisoning 24 hours after ingesting it. You want to find out which cheese it is, so you decide to use mice to taste them. But, you only have two mice left in your lab and your deadline to find the poisoned cheese is exactly 24 hours from now! You can assign every mouse the number of cheeses that you want. How would you do it?



I would assign mouse 1 to cheese/s _____

I would assign mouse 2 to cheese/s _____

SOLUTION: The two mice should share one cheese (e.g., cheese 2), each mouse should eat one different cheese (e.g., mouse 1 eats cheese 1 and mouse 2 eats cheese 3), and leave aside one cheese (e.g., cheese 4 in that example).

19

You have two ropes and a lighter, and each rope takes exactly one hour to burn. They don't burn randomly, but at a regular speed. How would you use them to time exactly 15 minutes? The ropes cannot be cut, bent, straightened or measured with external instruments. (You don't need to time 15 minutes from the moment that you start your actions)



Please, briefly explain the procedure:

SOLUTION:

1. Light rope 1 at both ends and rope 2 at one end.
2. When the two flames on rope 1 meet, 30 minutes will have passed. Rope 2 has 30 minutes left of burn-time.
3. At that point, light Rope 2 at the other end.
4. In exactly fifteen minutes, Rope 2 will be completely burnt

To score 1, it's important that the subject refers to procedures 1 and 3.

If the subject mentions other procedures to measure 15' that involve manipulating the ropes (put both ropes in a cross-shape, bend them, mark them with external tools, etc.) this is not correct and it should be scored 0.

3

You have been given 9 coins of the same value, but one of them is fake which you could tell because it is lighter than the rest. You have a balance like the one in the picture to weigh the coins, and each weighing can result in “the balance leans to the right”, “the balance leans to the left”, or “the balance rests stable”. Assuming you are following an optimal strategy to reduce the number of weighings, how many weighings are necessary to identify the fake coin?



SPACE LEFT BLANK for answer

SOLUTION: 2