Developing a Computational Thinking Test using Bebras problems

James Lockwood
Dept. of Computer Science
Maynooth University, Maynooth, Co. Kildare, Ireland
james.lockwood@mu.ie
Aidan Mooney
Dept. of Computer Science
Maynooth University, Maynooth, Co. Kildare, Ireland
aidan.mooney@mu.ie

Abstract

Assessment is one of the major factors to consider when developing a new course or program of study. When developing a course to teach Computer Science there are many forms this could take, one of which is linked to Computational Thinking. Whilst developing Computer Science to Go (CS2Go), an introductory course aimed at secondary school students, we have developed a Computational Thinking test based on the problems developed for the international Bebras Challenge.

This paper will describe the content and development of the course, as well as some analysis on results from a year-long study with secondary school students and first-year undergraduate students. We believe that, based on our analysis and previous research in the field, that our assessment, based on pre-exisiting Bebras problems, has the potential to offer educators another way of testing this increasingly discussed skill, Computational Thinking.

In: A. Piotrkowicz, R. Dent-Spargo, S. Dennerlein, I. Koren, P. Antoniou, P. Bailey, T. Treasure-Jones, I. Fronza, C. Pahl (eds.): Joint Proceedings of the CC-TEL 2018 and TACKLE 2018 Workshops, co-located with 13th European Conference on Technology Enhanced Learning (EC-TEL 2018), 03-09-2018, published at http://ceur-ws.org

1 Introduction

1.1 Computer Science to Go (CS2Go)

Computer Science to Go (CS2Go) is a course designed to teach Computer Science topics with a focus on Computational Thinking. The idea to develop a course arose from a need identified by our research group working with schools around Ireland, through the PACT programme. We observed that teachers were keenly interested in delivering Computer Science lessons and this led to more schools and teachers joining the programme. It has been our intention from the outset to expand the content on offer and to investigate what other topics and methods could be used [MDN+14].

Due to the fact that there is little in the way of a full course in Computational Thinking there was an opportunity and a desire to create a more complete and intensive course for Transition Year, with a view to developing it into a Junior Certificate short course. In Ireland the second level school system consists of an optional Transition Year (fourth year) which is one-year in length and is taken after the Junior Cycle (first to third year) and before the two-year Leaving Certificate programme, culminating in a final state exam. In September 2016, teachers who had previously been involved with our group, as well as others including trainee teachers, were asked for their ideas and inputs on course design and content. This feedback, in conjunction with input from our group members and an extensive literature review, led to the setting out of the following aims for a course which are presented in no particular order:

- Introduce students to Computer Science, what it is, how it can affect their lives, how they can be involved.

- Improve students CT and problem-solving skills by making them aware of a problem-solving process and how it can be beneficial in many subjects and areas of life.
- Improve students understanding of Computer Science including an imbalance in participation rates across genders and a stereotyped view of who engages in Computer Science.
- Teach students Computer Science concepts such as Algorithms, Cryptography, Sorting/Searching Algorithms etc. with a focus not just on the concepts themselves but on real-world applications.
- Teach students programming to some level.

Students who have participated in PACT courses in the past have commented that the modules had been both enjoyable and a good way to develop programming and other skills such as team work. However, they also stated a desire for more practical applications and we have been working to ensure that the topics and methods used in this course reflect their feedback ([MDN$^+$14]). The new course has since been designed and tested and has been well received by both students and teachers [LM18b].

1.2 Goals of the Test

Assessment is one of the key factors when designing and developing courses for any level of education. One of the areas that was needed to analyse the success and impact of CS2Go was to find or develop a Computational Thinking test. It had to fit the following requirements:

- Be applicable to the target age range (15-17 years old).
- Allow for differentiation between strong and weaker students (i.e. have harder and easier questions).
- Allow students to complete the questions without any prior knowledge.
- Be completed within a 40-minute class time.
- Allow for a pre- and post-test of similar difficulty and content.
- Test students Computational Thinking skills.

1.3 Computational Thinking

Denning [Den09] suggested that Computational Thinking (CT) has been around since the 1950s as algorithmic thinking, referring to the use of an ordered precise set of steps to solve a problem and where appropriate to use a computer to do this task. Seymour Papert [Pap80] is credited as concretising CT in 1980 but it is since the contribution of Jeanette Wing [Win06], who popularised the term and brought it to the international communitys attention, that more and more focus has been placed on CT within education. In her seminal paper, Wing outlined how she believed that all children should be taught CT placing it alongside reading, writing and arithmetic in terms of importance. She further described it as representing a "universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" [Win06].

Although academics have failed to agree on a universal definition of CT, Wing defines it as solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science. She states that it is not programming and that it means "more than being able to program a computer. It requires thinking at multiple levels of abstraction" [Win06]. In 2008 Wing posed a question to the computer science, learning sciences and education communities: "What are effective ways of learning (teaching) CT by (to) children? [Win08]. This in turn raised further questions about what concepts to teach, the order in which these might be taught, and which tools should be used to teach them.

In the meantime, a lot of work has been done around the world and across all levels of education to introduce CT into schools, colleges, after school clubs, mainly through Computer Science or computing classes/courses. As CT is important to a computer scientist this makes sense; however, it should be noted that being able to think computationally, which includes skills such as decomposition, abstraction, algorithmic thinking and pattern matching, can be of benefit to all disciplines. [Bun07] has made this point stating that CT concepts have been used in other disciplines and that the ability to think computationally is essential to every discipline.

A wide array of topics has been used to introduce CT to students. In addition to explicitly teaching students what CT is [GCP14, LHW16] students may be introduced to concepts such as abstraction [AD16, SS15], modelling [CN13], Algorithms [AD16, FLM$^+$15, MDN$^+$14], decomposition [AD16] and problem solving/critical thinking skills [RFP14, SS15].

1.4 CT Assessment

Assessment of CT is in it's infancy and as such, there aren't many methods for educators to test what is being described more as a central skill for students to possess.

Of note is one effort to develop a Computational Thinking test called the Computational Thinking Test (CTt) and another project called Dr. Scratch. Dr Scratch anaylses Scratch projects to deliver a CT score based on a number of different metrics [MLRG15]. This is a great tool and we recommend it as a tool to analyse Scratch projects developed in one module of CS2Go. As it works exclusively with Scratch, this didn't suit our purposes to study students "general" CT skills pre- and post-course. The CTt test has been developed as a series of multiple-choice questions that are presented online in either a "maze" or "canvas" interface. There a number of factors which define the questions [Gon15]. The group have analysed these two metrics (CTt and Dr Scratch) alongside the Bebras problems [RGMLR17]. They found that CTt was partially convergent with the other two and claim this is to be expected as the three assess CT but from different perspectives. They claim that CTt has a strength that it can be done in "pure pre-test conditions". This can allow early detection of problems but also doesn't allow for contextualised assessment. This is a strength of the Bebras problems, which has "real-life" questions but they also claim the "psychometric properties of some of the problems are still far off being demonstrated".

With this being said, we felt that, from assessing various forms of assessment for Computational Thinking that exist, both through a systematic literature review [LM18a] and through interactions with other researchers and educators it was decided to develop a test based on the Bebras competition problem for CS2Go.

1.5 Bebras Problems

Bebras is an international competition which aims to promote Computer Science and computational thinking among school students at all ages. Participants are usually supervised by teachers and the challenge is performed at schools using computers or mobile devices.

As part of their work in schools the PACT group are involved in the Irish version of this test and have designed and used Bebras problems in order to provide teachers with resources to introduce students to Computational Thinking. They are designed to be 3-minute-long questions and require no prior knowledge of programming or Computer Science topics. All the problems are linked to topics in Computing such as Cryptography, Trees etc. and this allows them to be used to introduce students to these topics without students even realising they are learning them.

The fact that the Bebras problems are designed to test Computational Thinking skills means they are well suited to test students Computational Thinking skills before and after the course. Gouws et al. [GBW13] previously used the South African version of Bebras in a similar manner and it was this that inspired the development of our own Computational Thinking test. Other studies have also been carried out on the Bebras problems to investigate both their effectiveness and to compare them to other Computational Thinking tests [HM15, Van14, DS16, HM14].

2 Methodology

The current format of the Bebras challenge doesn't suit as a comparative test as the questions change each year. The challenge is often conducted on PC's and we wanted to allow teachers to do it through either pen and paper or online if desired. It was decided that 13 questions would be used in each test, with students allowed 35 minutes to complete them. This considers both the 3-minute design of the question as well as the fact that some of the questions are designed for a younger age group than the target demographic. It was hoped that each test would be as close as possible to each other in terms of difficulty level as well as question topic and type. To do this many questions from Bebras challenges across the world were examined and critiqued.

The questions used in the UK challenges were deemed most appropriate and the contents of the test were sources from the 2015 and 2016 challenges. For the target age group (15-17-year olds) the UK challenge involves 18 multiple-choice questions over 40 minutes. As explained previously this was adjusted slightly for our purposes to be shorter but also allowed for some non-multiple-choice questions as well. The first criteria for the tests was to ensure that they were as close in terms of difficulty level as possible. The UK Bebras challenge is broken into six age groups as presented in Table 1.

Table 1: Bebras UK Sections

Group Name	Year Group	Approx. age
Kits	2 &3	6-8
Castors	4 & 5	8-10
Juniors	6 & 7	10-12
Intermediate	8 & 9	12-14
Seniors	10 & 11	14-16
Elites	12 & 13	16-18

Each age group is then further divided into three Sections, namely, Section A, Section B and Section C. Questions in Section A are considered the easiest with Section C problems being the more complex. Questions that are submitted for the Bebras problem are reviewed by a panel of experts in Computing education who are involved in the Bebras challenge. Questions that are accepted for either the qualification rounds, or the final challenge are often used in multiple age groups and across the three Sections.

To ensure that each created test was as similar in difficulty as possible these ratings were used to select questions for each test, ensuring that corresponding problems were used in at least one common section and age group. The chosen corresponding problems for the tests along with the sections they have in common can be seen in Table 2. For the complete set of problems consult `goo.gl/XDRHbq`.

Table 2: Matching sections of the tests

Test 1 Question	Common Sections	Test 2 Question
Bracelet	Kits B, Castors A	Bebras Painting
Animation	Castors B, Juniors A	Bottles
Animal Competition	Castors B, Intermediate A	Party guests
Cross Country	Intermediate A	Tube System
Stack computer	Senior B, Elite A	Pirate Hunters
Throw the dice	Juniors C	Magic potion
Drawing stars	Intermediate B	Concurrent directions
Beaver lunch	Senior B	Theatre
You won't find it	Intermediate C, Elite A	Secret messages
Bowl Factory	Intermediate C, Elite B	Triangles
Fireworks	Senior C	Scanner code
Kangaroo	Elite C	The Game
Spies	Elite C	B-enigma

The second criteria for the tests was to have similar topics and styles for the questions where possible, and to have these topics relating to areas covered in the course. This was not as much a priority as the difficulty, so questions were considered even if this wasnt possible. Table 3 presents the topics covered by each question for each test.

Table 3: Topics of the questions

Test 1	Topic	Test 2	Topic
Bracelet	Pattern Matching	Bebras Painting	Algorithms
Animation	Attributes and Variables	Bottles	Sorting
Animal Competition	Data ordering	Party guests	Graphs
Stack computer	Stacks	Pirate Hunters	Graphs
Throw the dice	If-then-else	Magic potion	Logic & binary
Drawing stars	Objects	Concurrent directions	Parallel instructions
Beaver lunch	Trees	Theatre	Sequences
You won't find it	Ciphering	Secret messages	Ciphering
Bowl Factory	Sorting	Triangles	Iterative, pattern matching
Fireworks	Encoding	Scanner code	Pixels
Spies	Gossip Problem	B-enigma	Encrypting

Prior to either of the tests being used, they were tested by a small group to ensure that the questions were clear, made sense, that our timing (35 minutes) was reasonable, and that both sets of questions appeared similar in terms of difficulty. The group found that the second test was perhaps slightly harder, but that for 35 minutes it was doable and that the questions were clear in general.

To further assess that the two tests are similar in difficulty and validate their effectiveness the questions were sent out to teachers, undergraduate and postgraduate students and third level academic staff with instructions of how to rate the questions difficulty. The hope was that this sample of different demographic and career groups would show not only that the two tests are similar in difficulty but allow us to weigh either specific questions or one of the tests accordingly if there was a discrepancy. Tables 4, 5 and 6 presents the qualifications and areas of work of the participants. There was a mixture of genders and ages but this data was not collected, this group will be referred to as the panel from now on.

We asked the panel to rank the questions for us on two scales. Twenty people completed this task for Test 1, with 18 of those also completing it for Test 2. The

Table 4: Qualification profile of the panel

Highest Qualification	No. of participants
PhD	5
Masters	1
Bachelors Degree	10
Leaving Certificate	3
Unspecified	1

Table 5: Job profile of the panel

Job Title	No. of participants
Lecturer	5
Primary school teacher	2
Secondary school teacher	1
Tutor/Postgraduate Student	5
Youth worker	2
Nurse/Veterinary Nurse	2
Undergraduate Student	2
Unspecified	1

Table 6: Area of work of the panel

Area of work	No. of participants
Computer Science	9
Irish	1
Mathematics	1
Electronic Engineering	1
Youth work	2
Medicine	2
Teaching	3
Unspecified	1

first scale was rating the questions in each tests from easiest to hardest, this gave each question a ranking from 1 to 13. To further enhance this ranking a second scale was needed, as some questions might be classified as being the easiest two, but there could be a big gap in difficulty between them. The same could be true of any two questions. Since each test had 13 questions it was decided that a scale from 1-10 wouldn't allow the panel to be clear and would in fact limit the ranking. A scale of 1-20 was decided upon, with 1 being easiest and 20 was hardest. The panel weren't given further instruction unless it was requested, they were free to rank the questions as they saw fit.

3 Results

When asked to rate the questions on a scale from 1-20, Table 7 presents the scores for each question. The questions are presented from easiest to hardest based on the average scores in the table.

Table 7: Rating of Test 1 Questions out of 20

Rank	Question	Average out of 20
1	Bracelet	3.2
2	Animation	5.6
3	Cross Country	6.15
4	Beaver Lunch	7.6
5	Drawing Stars	7.6
6	Throw the Dice	7.7
7	You Wont Find It	8.75
8	Animal Competition	8.95
9	Kangaroo	9.05
10	Fireworks	9.45
11	Bowl Factory	12.6
12	Stack Computer	13.5
13	Spies	15.1
	Average	8.87

There isn't much of a difference between this ranking and the ranking participants gave the questions out of 13. This is to be expected, and the two rankings are presented in Table 8 to show this comparison.

Table 8: Test 1 Comparison

Question 1-13	Rank	Question 1-20
Bracelet	1	Bracelet
Animation	2	Animation
Cross Country	3	Cross Country
Throw the Dice	4	Beaver Lunch
Drawing Stars	5	Drawing Stars
Beaver Lunch	6	Throw the Dice
You Wont Find It	7	You Wont Find It
Kangaroo	8	Animal Competition
Animal Competition	9	Kangaroo
Fireworks	10	Fireworks
Stack Computer	11	Bowl Factory
Bowl Factory	12	Stack Computer
Spies	13	Spies

It should be noted that *Beaver Lunch*, *Drawing Stars* and *Throw the Dice* were rated as having almost exactly the same level of difficulty, with scores of 7.6, 7.6 and 7.7 respectively (see Table 7) out of 20. It should also be noted that there is a large jump in difficulty from 10th to 11th position (*Fireworks* to *Bowl Factory*). *Kangaroo* and *Fireworks* are rated 9.05 and 9.45 respectively, but the scores then jump significantly to 12.6, 13.5 and 15.1 for *Bowl Factory*, *Stack Computer* and *Spies*. A similar gap can be seen when going from the first three questions to the 4th question. *Bracelet*, *Animation* and *Cross Country* are rated as 3.2, 5.6 and 6.15 respectively, which in of itself covers a broad range. The score then jumps up to 7.6 for *Beaver Lunch* and *Drawing Stars*.

This lines up roughly with the age categories questions were used in during the Bebras competition. Table 9 presents a comparison between these three orderings. For the original category and UK results we have used the percentage in the highest category they were entered in, which can be seen in the table.

If we use the rankings in each of these columns we can attempt to rank the questions across all three columns to give an overall ranking. For example, Bracelet was ranked 1 in Column 1 and 1 in Column 2, giving a score of 2 (if we simply add these numbers together). If scores are identical in any of the columns then they will be given the same score e.g. in Column 2 *Beaver Lunch* and *Drawing stars* have a score of 7.6 so theyll both be given a value of 5 (i.e. the highest ranked question of the two).

In Column 3 a score will be given relating to the position of the highest question; for example, *You Won't Find It*, *Stack Computer* and *Drawing Stars* were all used in Elite A Category, so they will all be given a value of 10 as *Drawing Stars* is the highest placed in the list.

Doing this for each question we can then rank them from 1 to 13, with 1 being the easiest question (the lowest total across all four columns) and 13 being the hardest (the largest total across all four columns). This ranking is shown in Table 10.

The rankings shown in Table 10 can allow us to weigh questions by awarding a higher mark for getting a correct answer on harder questions and lower marks for a correct answer on easier questions. This hasn't been deemed necessary at this stage but further results might lead us to do this especially with hard questions such as *Spies*.

What we can deduce from this rankings is that the questions can be split into 3 different difficulty levels. Questions in Rank 1-4 all have a score of less than 15, they can be seen as the easiest four questions. Questions 5-10 have a ranking of between 15-30 and can be seen as intermediate questions and Questions 11-13 have rankings of over 30, and they can be seen as the hardest questions.

3.1 Test 2

When asked to rate the questions on a scale from 1-20, Table 11 presents the scores for each question as rated by the testers. The questions are presented from easiest to hardest. Unlike in Test 1 there appears to be more of a difference between this ranking and the

Table 9: Test 1 Extensive Ranking

Rank	Col 1 Ranking from 1-13	Col 2 Scores from 1-20	Col 3 Bebras Category (Highest)
1	Bracelet (1.75)	Bracelet (3.2)	Bracelet (Inter A)
2	Animation (4.8)	Animation (5.6)	Animation (Inter A)
3	Cross Country (4.95)	Cross Country (6.15)	Animal Competition (Inter A)
4	Throw the Dice (5.7)	Beaver Lunch (7.6)	Cross Country (Inter A)
5	Drawing Stars (6.3)	Drawing Stars (7.6)	Beavers Lunch (Senior B)
6	Beaver Lunch (6.65)	Throw the Dice (7.7)	Throw the Dice (Senior B)
7	You Wont Find It (7.15)	You Wont Find It (8.75)	Fireworks (Senior C)
8	Kangaroo (7.45)	Animal Competition (8.95)	You Wont Find It (Elite A)
9	Animal Competition (7.6)	Kangaroo (9.05)	Stack Computer (Elite A)
10	Fireworks (7.95)	Fireworks (9.45)	Drawing Stars (Elite A)
11	Stack Computer (9.6)	Bowl Factory (12.6)	Bowl Factory (Elite B)
12	Bowl Factory (9.85)	Stack Computer (13.5)	Kangaroo (Elite C)
13	Spies (11.25)	Spies (15.1)	Spies (Elite C)/

Table 10: Test 1 Ranking

Rank	Question	Total Score	Breakdown*
1	Bracelet	6	1+1+4
2	Animation	8	2+2+4
3	Cross Country	10	3+3+4
4	Throw the Dice	14	4+6+6
5	Beaver Lunch	17	6+5+6
6	Drawing Stars	20	5+5+10
7	Animal Competition	21	9+8+4
8	You Won't Find It	24	7+7+10
9	Fireworks	27	10+10+7
10	Kangaroo	30	8+9+13
11	Stack Computer	33	11+12+10
	Bowl Factory	33	12+11+10
13	Spies	39	13+13+13

*Rank of (col1 + col2 + col3 from) from Table 9

ratings given from 1-13, as shown in Table 12.

Table 11: Rating of Test 2 Questions out of 20

Rank	Question	Average out of 20
1	Bebras Painting	3.72
2	Tube System	5.94
3	Concurrent Directions	7.72
4	Magic Potion	8
5	Theatre	8.06
6	Bottles	8.44
7	Party Guest	8.44
8	Secret Messages	8.78
9	Triangles	10.89
10	Scanner Code	11.56
11	B-Enigma	11.78
12	The Game	11.78
13	Pirate Hunters	13.44
	Average	9.12

It is interesting to note that with this test it appears there are a few more discrepancies between the two rankings. From questions 3-11 there are several questions "out of place", some just one rank (like *Theatre* and *Bottles* in ranks five and six) or in the case of *Concurrent Directions* and *Party Guest*, differ by three or four ranks. The reason for this is that these questions were all rated very similarly by most people. In terms of the difficulty score (from 1-20) there is only one point separating *Concurrent Directions (7.72)* in third position, and *Secret Messages (8.78)* in eighth

Table 12: Test 2 Comparison

Question 1-13	Rank	Question 1-20
Bebras Painting	1	Bebras Painting
Tube System	2	Tube System
Magic Potion	3	Concurrent Directions
Party Guest	4	Magic Potion
Bottles	5	Theatre
Theatre	6	Bottles
Concurrent Directions	7	Party Guest
Secret Messages	8	Secret Messages
Scanner Code	9	Triangles
B-Enigma	10	Scanner Code
Triangles	11	B-Enigma
The Game	12	The Game
Pirate Hunters	13	Pirate Hunters

position. This is what leads to the slight mismatch in those positions. Similarly, there is only one point separating *Triangles (10.89)* in ninth place with *The Game (11.78)* in 12th place. Also, of interest is the fact that *Bottles* and *Party Guest* (both 8.44) and *The Game* and *B-Enigma* (both 11.78) were rated with the same level of difficulty.

These rankings line up roughly with the age categories questions were used in during the Bebras competition. Table 13 presents a comparison between these three orderings. For the original category and UK results we have used the percentage in the highest category they were entered in, which can be seen in the table.

If we use the rankings in each of these columns we can rank the questions across all three columns to give an overall ranking. For example, Bebras Painting was ranking 1 in Column 1 and 1 in Column 2, giving a score of 2, when added together. If scores are identical in any of the columns then they will be given the same score e.g. in column two B-engima and The Game have the same score, so theyll both be given a value of 12 (i.e. the highest ranked question of the two).

In Column 3 the score will be given of the highest question, for example Theatre, Scanner Code and Triangles were all used in Elite B, so they will all be given a value of 11 as Triangles is the highest placed in the list.

Doing this for each question we can then rank them from 1 to 13, with 1 being the easiest question (the lowest total across all four columns) and 13 being the hardest (the largest total across all four columns). This ranking is shown in Table 14.

As with Test 1 the rankings shown in Table 10 can allow us to weigh questions by awarding a higher mark

Table 13: Test 2 Extensive Ranking

	Column 1	Column 2	Column 3
Rank	Our Analysis (Ranking from 1-13)	Our Analysis (scores from 1-20)	Bebras Category (Highest)
1	Bebras Painting (2.22)	Bebras Painting (3.72)	Bebras Painting (Castors A)
2	Tube System (4.33)	Tube System (5.94)	Bottles (Junior A)
3	Magic Potion (5.39)	Concurrent Directions (7.72)	Party Guests (Inter A)
4	Party Guest (5.78)	Magic Potion (8)	Tube System (Inter A)
5	Bottles (6.22)	Theatre (8.06)	Concurrent Directions (Senior A)
6	Theatre (6.5)	Bottles (8.44)	Pirate Hunters (Elite A)
7	Concurrent Directions (6.78)	Party Guest (8.44)	Magic Potion (Elite A)
8	Secret Messages (7)	Secret Messages (8.78)	Secret Messages (Elite A)
9	Scanner Code (8.89)	Triangles (10.89)	Theatre (Elite B)
10	B-enigma (8.94)	Scanner Code (11.56)	Scanner Code (Elite B)
11	Triangles (9)	B-enigma (11.78)	Triangles (Elite B)
12	The Game (9.89)	The Game (11.78)	The Game (Elite C)
13	Pirate Hunters (9.94)	Pirate Hunters (13.44)	B-enigma (Elite C)

Table 14: Test 2 Ranking

Rank	Question	Total Score	Breakdown*
1	Bebras Painting	3	1+1+1
2	Tube System	8	2+2+4
3	Bottles	14	5+7+2
4	Party Guest	15	4+7+4
	Magic Potion	15	3+4+8
	Concurrent Directions	15	7+3+5
7	Theatre	22	6+5+11
8	Secret Messages	24	8+8+8
9	Scanner Code	30	9+10+11
10	Triangles	31	11+9+11
11	Pirate Hunters	34	13+13+8
12	B-enigma	35	10+12+13
13	The Game	37	12+12+13

*Rank of (col1 + col2 + col3) from Table 13

for getting a correct answer on harder questions and lower marks for correct answer on easier questions. This hasn't been deemed necessary at this stage but further results might lead us to do this with especially hard questions such as *The Game*.

Similarly to Test 1 we can deduce from this rankings that the questions can be split into 3 different difficulty levels. Questions in Rank 1-3 all have a score of less than 15, they can be seen as the easiest four questions. Questions 4-9 have a ranking between 15-30 and can be seen as intermediate questions and Questions 10-13 have rankings of over 30, they can be seen as the hardest questions. These divisions are similar to the divisions shown in Test 1.

4 Findings

Using the average of the difficulty of the two tests as presented in Table 7 and Table 11 it can be seen that both tests are of a similar difficulty level. Test 1 questions were rated on average at a perceived difficulty of 8.87 and Test 2 questions were rated on average at a perceived difficulty of 9.12. This leads us to be able to conclude that the two tests have a similar difficulty rating.

These tests have been run over the course of the 2017-18 academic year in a number of schools as well as on a first year undergraduate CS course. It was run in schools as part of the wider CS2Go roll-out and it was decided that running it with the undergraduate students would be helpful as they are a larger, more consistent sample. It was also felt that students could benefit from the problem solving aspect of the assessment.

With both cohorts it was hoped to see if the test could be completed in the 35 minute time-period allotted. It was also hoped that it could be seen from results that the test targets the students Computational Thinking skills. This is hard to really define but we used students previous mathematics and programming experience as metrics to compare groups. Mathematical ability has been shown to be a predictor of success in programming [QBM15] and most would agree that programming is a specific way of testing CT skills.

4.1 Overall Results

4.1.1 School data

A total of 200 took at least one of the problem-solving tests. Of those 200 students, 187 took Test 1 and 76 took Test 2. The decrease in number is due to some schools not completing all of the feedback and assessment at the end of the year. This could have been due to their teachers not using the content much or not having time to re-test the students.

Table 15 shows the results of the tests grouped by those to two at least one test and those who took both tests. It can be seen that students in both groups performed slightly better in the second test than the first test. For the whole population this is to a significant level (T-score = 2.473, P-value = 0.014) but for those who took both tests there is no significant difference (T-score = 0.159, P-value = 0.873).

Table 15: School results of the tests, where n is the number of students taking the test

	Average of those who took at least one test	Average of those who took both tests
Test 1	5.806 (n = 187)	6.527 (n = 55)
Test 2	6.627 (n = 76)	6.6 (n = 55)

These scores are all out of 13

4.1.2 Undergraduate Students

A total of 292 students took at least one of the problem-solving tests. Of those 292 students, 263 took Test 1 and 180 took Test 2. The decrease in numbers is due to students changing course, only needing to complete one semester of CS and other unrelated circumstances.

Table 16 shows the results of the tests grouped by those to two at least one test and those who took both tests. It can be seen that a total of 174 took both tests. Students performed marginally better in Test 2 compared to Test 1, but this isn't a significant difference (T-score = 0.129, P-value = 0.897). This increase is also found across the whole population with the averages being 7.689 for Test 1 (n=263) and 7.933 for Test 2 (n=180) (T-score = 1.17, P-value = 0.24).

Table 16: Undergraduate results of the tests, where n is the number of students taking the test

	Average of those who took at least one test	Average of those who took both tests
Test 1	7.689 (n = 263)	7.988 (n = 174)
Test 2	7.933 (n=180)	8.03 (n = 174)

These scores are all out of 13

As stated in Section 1.2, one of the hopes of these studies was to show that the Bebras problems challenge students Computational Thinking skills. For this we have looked at students who had previous programming experience and those who took Higher Level Mathematics at the Leaving Certificate.

Table 17 shows that students who took Higher Level maths performed significantly better in both Test 1 (T-score = 2.768 P-value=0.006) and Test 2 (T-score = 3.409 P-value = 0.001). This is encouraging as mathematical ability and Computational Thinking can be seen as closely related skill-sets. Also interestingly those who studied Ordinary Level Mathematics performed slightly worse in Test 2 than Test 1, whereas those who studied Higher Level increased slightly. It should be noted that neither groups scores changed significantly over the two tests.

It can also be seen that those who had previous programming experience performed better in Test 1 than those who had no experience, but not to a significant level (T-score = 0.853 P-value = 0.395). Interestingly, not only was the numerical gap closed in this demographic by Test 2, but had swung the other way, with those who had no previous experience out-performing their peers, although neither groups scores changed to a significant level. This is encouraging as it could help to show that the content covered by introductory CS courses, namely programming and low-level theory, are beneficial for Computational Thinking skills.

One way which we can compare the two difference cohorts is by looking at the percentage of students who got each question right. We would expect the undergraduate students to perform better in most, if not all, questions across both Tests.

4.2 Bebras Test 1

It can be seen from Table 18 that in Test 1 the undergraduate students performed better than those in secondary school in almost every question. Many of the ranges in scores are from 15-30%

Table 17: Undergraduate Demographic comparisons

Demographic	Test 1 Avg	Test 2 Avg
Studied OL (n = 44)	7.386	7.273
Studied HL (n = 110)	8.445	8.509
PP (n = 71)	8.225	8.085
NPP (n = 88)	7.92	8.102

OL = Ordinary Level Mathematics
HL = Higher Level Mathematics
PP = Previous Programming Experience
NPP = No previous programming experience

Table 18: Our Results Test 1

Question	Undergrads (n=277)	Schools (n=186)
Bracelet	94.6%	94.1%
Animation	70.8%	56.5%
Animal Competition	64.9%	39.8%
Cross Country	68.6%	45.2%
Stack Computer	44%	13.9%
Throw the Dice	77.9%	50.5%
Drawing stars	82.3%	60.2%
Beaver Lunch	33.2%	29%
You wont find it	89.2%	74.2%
Bowl Factory	18.4%	11.8%
Fireworks	42.2%	47.8%
Kangaroo	56.7%	40.3%
Spies	29.6%	17.2%

between the two groups. This is to be expected as the students have been through at least two more years of education and one of the expressed goals of the Leaving Certificate is to develop students into critical and creative thinker thinkers (`https://www.curriculumonline.ie/getmedia/161b0ee4-706c-4a7a-9f5e-7c95669c629f/KS_Framework.pdf`), which is all connected to Computational Thinking. The only question where the secondary school students out-performed the undergraduates was the *Fireworks* question.

4.3 Bebras Test 2

From Table 19 we can see that, like in Test 1, the undergraduate students performed better than the secondary school students in Test 2. The gaps this time are generally lower though, with most being around 10%. The secondary school students again performed better in one question, *The Game*. This is interesting as, based on the percentage of students who got the question right, and our own analysis discussed in Section 3, The Game was identified as being one of the hardest questions across both tests.

Table 19: Our Results Test 2

Question	Undergrad (n=197)	School (n=75)
Bebras Painting	81.7%	58.7%
Bottles	90.4%	84%
Party Guests	88.3%	70.7%
Tube System	67.5%	57.3%
Pirate Hunters	42.1%	33.3%
Magic Potion	83.2%	69.3%
Concurrent Directions	78.7%	66.7%
Theatre	32.5%	21.3%
Secret Messages	86.8%	86.7%
Triangles	47.2%	34.7%
Scanner Code	36.5%	32%
The Game	3.6%	5.3%
B-Enigma	51.3%	42.7%

5 Conclusions

Based on the analysis from our panel we can conclude that the two tests are of approximately equal difficulty. As discussed in Section 1.4 and 1.5 the Bebras problems have been developed to test participants CT skills and this compares well to other existing tests like the CT. This is further backed up by our findings from the undergraduate students as those who had previously programmed and who studied Higher Level mathematics achieved higher results in Test 1.

One advantage of this test is it's ability to be administered both online or through paper question and answer sheets. However, it is clear that with technology use in schools becoming more commonplace, online submission is preferable. This is also true from a data collection point of view, as it can save time and effort as well as provide almost immediate results. The results presented here were collected via both paper answer sheets as well as using Google forms to collect online responses. This has worked as a stop-gap but a more robust and controlled system is needed. To that end a web-system for the entire CS2Go course, as well as the assessment tools described here, has been developed over the past year. It will go live this summer and it is hoped this will allow easier access for both educators and our research group to data and course content.

One interesting development that we plan to pursue would be to develop "equivalent" Bebras problems. Each Bebras exercise is usually based around a specific CS-related concept or problem. To not only make the test equivalent in difficulty but also topic, we would have to develop Bebras exercise which have the same underlying concept or idea but with a different story or real-world application. This is no easy task but if a method could be developed this would not only help our test but also allow the Bebras challenge itself to develop similar questions year after year.

An area of interest in our research group is methods of predicting success in programming courses. Being able to implement interventions to help students seen as potential struggling students is vitally important and beneficial to all educators. If this test could be shown to predict success in either programming or general academic success it could be a helpful tool for educators. We plan to use the data obtained from the undergraduate students and their final grades to begin to see if this is possible.

6 Acknowledgements

Thank you to all of those who submitted rankings.

References

[AD16] S. Atmatzidou and S. Demetriadis. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75:661–670, 2016.

[Bun07] A. Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2):67–69, 2007.

[CN13] M. E. Caspersen and P. Nowack. Computational thinking and practice: A generic approach to computing in danish high schools. In *Proceedings of the 15hth Australasian Computing Education Conference*, pages 137–143, January 2013.

[Den09] Peter J Denning. The profession of it beyond computational thinking. *Communications of the ACM*, 52(6):28–30, 2009.

[DS16] V. Dagiene and G. Stupuriene. Informatics in education. In *Bebras-a sustainable community building model for the concept based learning of informatics and computational thinking.*, 2016.

[FLM+15] R. Folk, G. Lee, A. Michalenko, A. Peel, and E. Pontelli. Gk-12 dissect: Incorporating computational thinking with k-12 science without computer access. In *Proccedings Frontiers in Education Conference (FIE)*, October 2015.

[GBW13] L. Gouws, K. Bradshaw, and P. Wentworth. October. In *First year student performance in a test for computational thinking*, pages 271–277. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference . ACM, 2013.

[GCP14] S. Grover, S. Cooper, and R. Pea. Assessing computational learning in k-12. In *Proceedings of Conference on Innovation & technology in computer science education*, pages 57–62, June 2014.

[Gon15] M. R. Gonzlez. Edulearn15. In *Computational thinking test: Design guidelines and content validation.*, 2015.

[HM14] P. Hubwieser and A. Mhling. 9th workshop in primary and secondary computing education (wipsce). In *Playing PISA with bebras.*, 2014.

[HM15] P. Hubwieser and A. Mhling. Learning and teaching in computing and engineering (latice). In *Investigating the psychometric structure of Bebras contest: towards mesuring computational thinking skills.*, 2015.

[LHW16] W. L. Li, C. F. Hu, and C. C. Wu. Teaching high school students computational thinking with hands-on activities. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 371–371, July 2016.

[LM18a] J. Lockwood and A. Mooney. Computational thinking in secondary education: Where does it fit? a systematic literary review. *International Journal of Computer Science Education in Schools*, 2018:41–60, January 2018.

[LM18b] J. Lockwood and A. Mooney. A pilot study investigating the introduction of a computer-science course focusing on computational thinking at second level. *The Irish Journal of Education/Iris Eireannach an Oideachais*, Forthcoming 2018.

[MDN+14] A. Mooney, J. Duffin, T. Naughton, R. Monahan, J. Power, and P. Maguire. Pact: An initiative to introduce computational thinking to second-level education in ireland. In *Proceedings of Interna-*

tional Conference on Engaging Pedagogy (ICEP), 2014.

[MLRG15] Robles G. Moreno-Len, J. and M. Romn-Gonzlez. Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educacin a Distancia*, 2015.

[Pap80] Seymour Papert. *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc., 1980.

[QBM15] K. Quille, S. Bergin, and A. Mooney. Press#, a web-based educational system to predict programming performance. *International Journal of Computer Science and Software Engineering*, pages 178–189, 2015.

[RFP14] J. F. Roscoe, S. Fearn, and E. Posey. Teaching computational thinking by playing games and building robots. In *Proceedings International Interactive Technologies and Games Conference (iTAG)*, October 2014.

[RGMLR17] 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.

[SS15] J. Shailaja and R. Sridaran. Computational thinking the intellectual thinking for the 21st century. *International Journal of Advanced Networking & Applications Special Issue*, 2015:39–46, May 2015.

[Van14] Jiří Vaníček. Bebras informatics contest: criteria for good tasks revised. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pages 17–28. Springer, 2014.

[Win06] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

[Win08] 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.