The Mathability of Computer Problem Solving with ProgCont

Piroska Biró[1], Tamás Kádek[2]

[1]University of Debrecen, Faculty of Informatics, Kassai út 26, 4028 Debrecen, Hungary, biro.piroska@inf.unideb.hu

[1]Sapientia Hungarian University of Transylvania, Faculty of Economics, Socio-Human Sciences and Engineering, Miercurea-Ciuc, Romania, biropiroska@uni.sapientia.ro

[2]University of Debrecen, Faculty of Informatics, Kassai út 26, 4028 Debrecen, Hungary, kadek.tamas@inf.unideb.hu

Abstract: In the teaching/training of programmers, the development of mathematical skills is given as much priority as that of IT skills. A complex IT problem solution is inconceivable without adequate mathematical background knowledge. In our research, we would like to show this connection through the analysis of programming tasks, for which we use the automatic solution evaluation system developed by the Faculty of Informatics of the University of Debrecen almost a decade ago and the data accumulated during its use. We examine the effectiveness of users in solving programming tasks that require mathematical and IT knowledge, compare performance in different programming languages, and look for topics where improvement is needed. The results show that, contrary to our expectations, students perform better on tasks that require both mathematics and informatics skills.

Keywords: ProgCont system; automated evaluation; programming; mathability; computer problem solving

1 Introduction

At the Faculty of Informatics of the University of Debrecen, teaching mathematics and informatics subjects goes hand in hand with the various IT majors (Computer Science, Computer Science Engineering, Business Informatics). Knowledge in the relevant fields of both disciplines is essential for the successful completion of these majors. The two disciplines are also often intertwined within subjects. Subjects aimed at acquiring programming skills, from subjects introducing algorithmic thinking to subjects describing high-level programming languages, are no exception in this respect [12], [20]. In teaching different programming languages, program writing tasks are typically used to measure students' knowledge. In the case of these tasks, both mathematical and IT knowledge are required, albeit to a different extent, depending on the type of the task. In our research, we would like to understand the relationships and differences between these types of tasks by examining the effectiveness of our students in solving tasks [11], [19].

One of the basic goals of the ProgCont system developed at the University of Debrecen, Faculty of Informatics since 2011, is to develop students' mathematical and IT skills and to promote students' learning and practice. The system primarily assists in the teaching of programming languages by testing and evaluating programs submitted in different programming languages [13]-[17].

Based on nearly a decade of experience and use, we can say that such a system and tool has been developed that also improves students' mathability skills [4]-[10], [18]. The concept of mathability [2] was born under Cognitive Infocommunications [1], [3].

In this article, we compare student performance measured on tasks requiring mathematics and informatics skills over seven years. In our research, we use three hypotheses to look for correlations between these abilities.

2 Research

2.1 The ProgCont System

The ProgCont system has been developed and used by the Faculty of Informatics from 2011 to the present day for automatic evaluation of programming tasks [13]-[17]. The software is very similar to the Moodle CodeRunner plugin. It compiles, runs, and checks solutions for programming tasks submitted in various source languages (C, C++, C#, Java, Pascal, Python), comparing their output with the expected correct output.

The system was developed primarily for programming competitions to objectively evaluate competition tasks. However, it soon became clear that it could also be used in examinations and preparing for examinations. Over the last decade, the system has been growing steadily. More and more different tasks have been formulated, and it has become possible to support more and more programming languages. Up to now 45 competition problem sets, 241 examination problem sets, 11 practice problem sets are available in the system with a total of 1 657 tasks.

At first glance, the number of practice problem sets may seem remarkably low compared to the examination tasks, however, the examination tasks remain available for practice after the examination. In addition to the usual programming languages (C, C++, Java, Pascal) in international ACM-style programming competitions, support for the C# programming language soon appeared, and the list has been recently expanded to include Python.

Table 1
Table of all submissions by programming languages until 31/08/2020

Programming language	**Submissions**	**Accepted**	**Pass rate**
ANSI C	68 201	19 257	28 %
C (C99)	118 992	40 925	34 %
C++	9 983	2 752	28 %
C#	19 892	8 703	44 %
Java	58 258	22 237	38 %
Pascal	1 366	150	11 %
Python	418	99	24 %
Total	**277 110**	**94 123**	**34 %**

Due to the diverse nature of the use, the range of users is also very wide:

- high school students preparing for, or participating in, competitions;
- participants of extracurricular activities for high school students organized by our faculty;
- IT students of our faculty preparing for, or participating in, competitions;
- students preparing for and taking examinations in, certain subjects.

The ProgCont system has been developed primarily based on non-pedagogical aspects, but at the same time it is playing an increasingly important role in supporting education, and in the last 10 years so much data has been collected that should be analysed with pedagogical methods.

2.2 Hypotheses

The present research aims to use the database, which has been growing for almost a decade, to examine and compare the IT and mathematical skills of the widest possible range of users and the possible connection between them. The following hypotheses were defined:

Hypothesis 1: *Users of our system perform better in solving tasks that require only IT skills than those where both IT and mathematical competencies are required.*

Hypothesis 2: *A set of publicly available tasks in the ProgCont system can be identified where users perform poorly (we can suggest types of problems where more tasks should be set up to improve user performance).*

Hypothesis 3: *There exists a relationship between the type of task and the programming language chosen for its solution that can be observed among our users.*

2.2.1 SWOT Analysis

Strengths

A database is available that contains a significant number of tasks (1 657) and objective (automatically performed) evaluation of a large number (277 108) of the submitted solutions.

Weaknesses

Task solutions cannot be bound to individuals. We have deliberately kept the ProgCont system, especially since the release of GDPR, away from storing information that requires the processing of personal data. Thus, for example, the student whom a solution submitted for evaluation belongs to during an assessment is managed by an external system, and thus, no information is available during the research. We can only make statements that are generally valid for ProgCont users.

Opportunities

The pandemic has given a huge boost to the development of online education and has, in the eyes of many, appreciated the potential that resides in our ProgCont software. We can certainly hope that the number of users will increase further shortly, which will provide an opportunity to expand the present research in the future. Therefore, in the analyses examining the hypotheses discussed in this article, we have observed that this can be easily repeated in the future.

Threats

By adopting the provisions of GDPR, the change in the legal environment has not stopped. We often find that some institutions, including our university, transpose regulation into their practice in a stricter way. This requires constant adaptation.

2.2.2 The Research Sample

To test the hypotheses on as large a sample as possible, we chose tasks that have been available for some time and may have been of interest to a wide range of users. Therefore, we chose a collection of nearly 100 tasks that had already been collected in 2014 in the ProgCont system as the basis for our research Table 2.

The advantage of the collection is that the tasks are grouped based on 9 topics related to informatics or mathematics. As in ProgCont most of the problem sets contain previously published problems, we also examined evaluated solutions coming from the time before the collection was compiled. Due to all this, we chose seven years for the analysis, from 01/12/2012 to 01/12/2019. For all examined tasks, at least one solution was received before 01/12/2012, so all tasks were available to users during the examined period.

Since the aim is to compare the performance of solving tasks requiring IT and mathematical skills, we kept only those 3-3 of the 9 topics for which these skills can be well distinguished. We selected 3 informatics and 3 mathematics topics area (Table 2).

Category of mathematics tasks:

- Number theory (7 tasks)
- Geometry (5 tasks)
- Arithmetic and algebra (12 tasks)

Category of informatics tasks:

- Simulation (10 tasks)
- String operations (12 tasks)
- Sorting and searching (6 tasks)

Table 2
The sample and the categories

Category	**Submissions**	**Accepted**	**Pass rate**
Simulation	585	91	16 %
String operations	1 100	191	17 %
Sorting and searching	703	167	24 %
Informatics	**2 388**	**449**	**19** %
Number theory	1 553	334	22 %
Geometry	371	83	22 %
Arithmetic and algebra	1 505	625	42 %
Mathematics	**3 429**	**1 042**	**30** %
Total	**5 817**	**1 491**	**26** %

In order not to distort the results, for example, the examination of tasks requiring knowledge of graph algorithms was omitted from the processing, as these included tasks that could be classified in both the former and the latter category or were difficult to classify. Tasks with a remarkably low number of submitted solutions (less than 20) were not processed either.

In the indicated period, a total of 5 817 submissions were received for the selected tasks, and 1 491 of them were correct, which means 26% of the total.

2.2.3 The Programming Task Categories

Based on Figure 1, it can be seen that the first hypothesis is not satisfied. There is a significant difference between the two groups, so the pass rate is higher for mathematics tasks than for informatics tasks (Figure 1 and Figure 2).

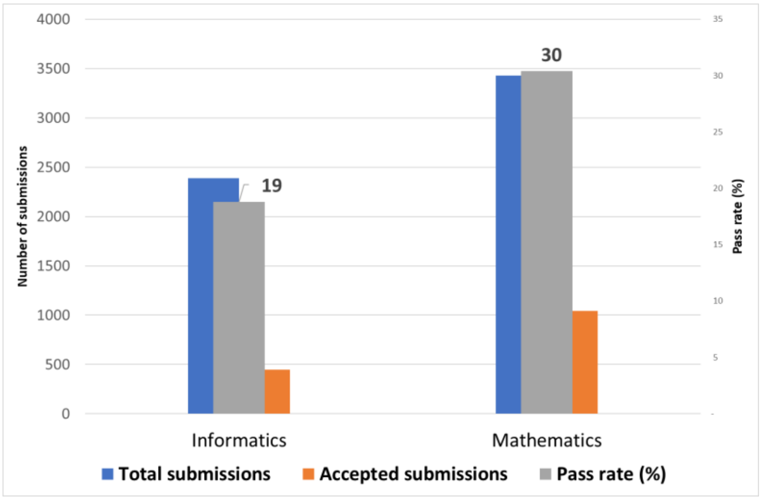

Figure 1
The acceptance rate in the two disciplines

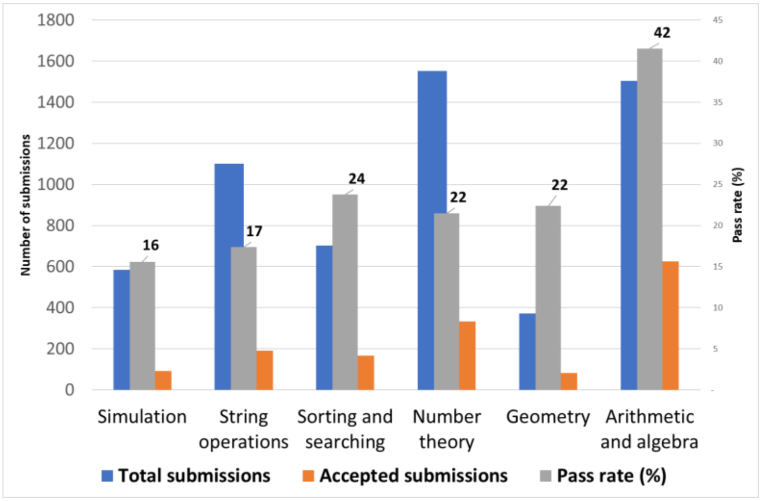

Figure 2
The acceptance rate in the selected categories

Students may be more confident in their knowledge of mathematics during programming learning than in their newly acquired IT skills; that is why submissions for tasks classified in the mathematics category are more successful. Greater emphasis needs to be placed on improving IT skills. To examine this in more detail, it is also worth looking at the distribution of submissions by tasks.

2.2.4 Error Distribution of Submitted Source Codes

The ProgCont system can provide six possible outputs when evaluating submissions, which can be used to categorize submissions with errors:

– *Compile error:* the submitted solution contains a syntax error, so the source code cannot be compiled.

– *Wrong answer:* the submitted solution produced incorrect output.

– *Presentation error:* the submitted solution produced incorrect output, but this differs from the expected output only in whitespace characters.

– *Runtime error:* An error occurred while running the submitted solution.

– *Time out:* the submitted solution did not run within the specified time limit.

In the research, we did not address solutions rejected with the compile error message, as these cannot be evaluated. The presentation error feedback has been in use since May 2017, before which such errors also resulted in wrong-answer feedback. Therefore, in our present research, we have merged the two categories and included them under the wrong answer tag.

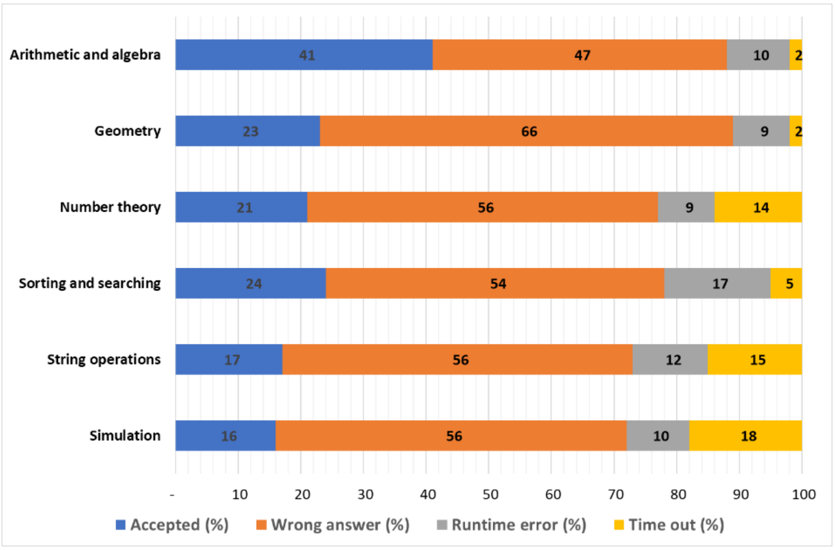

Figure 3

Distribution of programming errors in each programming task category

Figure 3 clearly shows that the most common type of error in submissions is the wrong answer in each category. It can also be seen from the figure that our users perform significantly worse in the case of tasks classified in the "String operations" and "Simulation" categories. Therefore, (confirming Hypothesis 2), it is necessary to expand the collection of tasks for these topics.

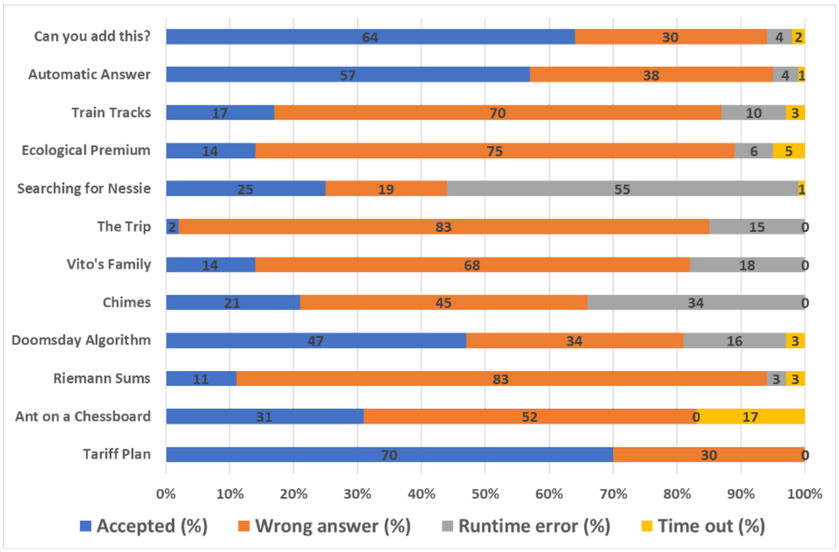

Figure 4

Distribution of programming errors by tasks in arithmetic and algebra

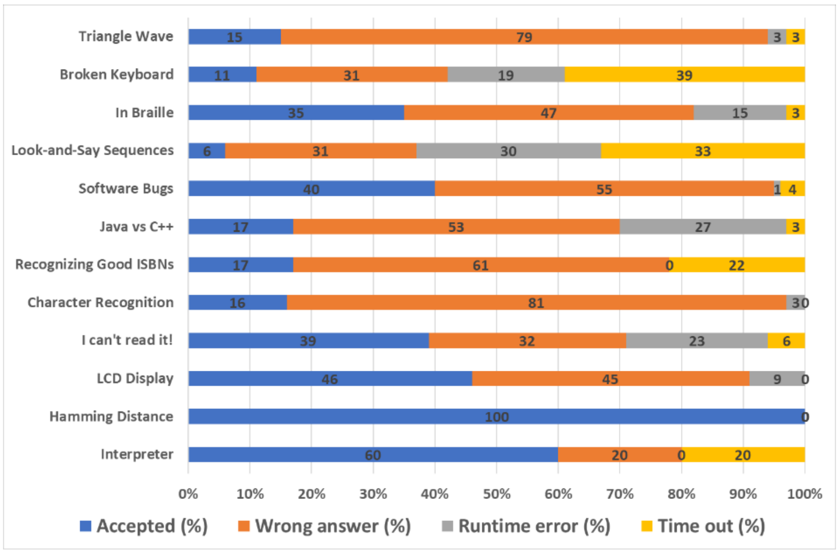

Figure 5

Distribution of programming errors by tasks in string operations

The difference between the typical errors in solving mathematical and IT problems is well shown by the comparison of task types “Arithmetic and algebra” and “String operations” with 12-12 tasks. The tasks were given in descending order of the number of solutions submitted. The four tasks “Tariff Plan”, “LCD Display”, “Hamming Distance”, and “Interpreter” are among the tasks that were omitted from the previous processing (less than 20 submissions). These are shown here (Figure 4 and Figure 5) only for the sake of completeness.

In addition to the overthrow of the first hypothesis, the number of occurrences of different error messages also shows surprising results. We thought that the “Time out” type of error would be more typical for mathematical problems. This is because a “Time out” error can typically be caused by someone trying to solve the problem as a simulation rather than using the appropriate solution formula. In light of this, it is quite surprising that this error is much more common for string operations than for arithmetic problems. It is difficult to distinguish between a wrong answer and a runtime error, for example, mistyping a single index variable in an array reference can result in either, depending on the specific case. Although we can conclude that the programs that correctly implement an incorrect algorithm fall into the wrong answer category. As can be seen from Figure 4 and Figure 5 this is more typical of mathematical problems.

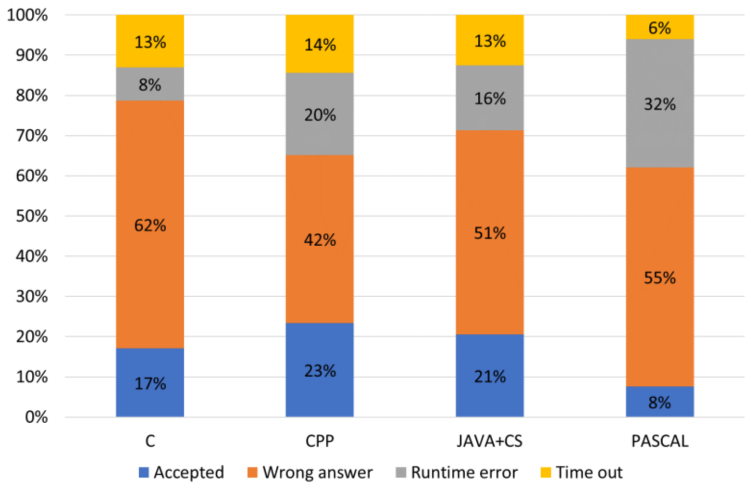

Figure 6

Distribution of programming errors by programming languages in tasks related to Informatics

Figure 6 and Figure 7 show the distribution of programming errors by the programming language in the two categories (informatics and mathematics). There were only a few submissions in the Python language (23 out of 5817) therefore we do not consider them. The comparison shows that students are more successful in solving informatics problems in an object-oriented language, while students are more successful in solving mathematics problems in C (Figure 6 and Figure 7).

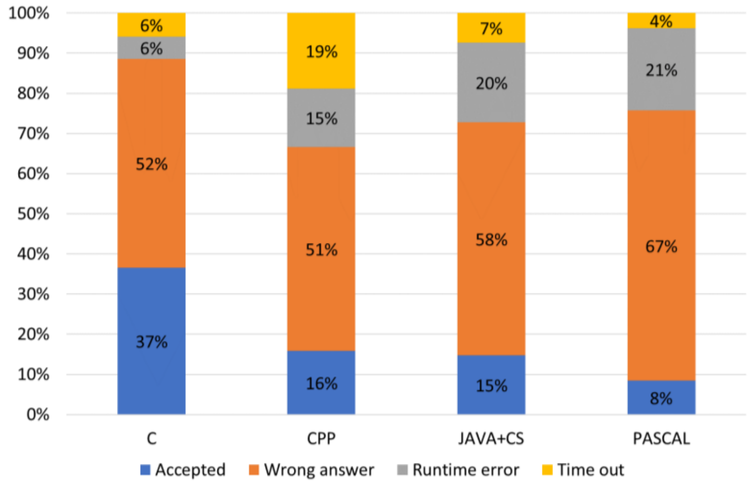

Figure 7
Distribution of programming errors by programming languages in tasks related to Mathematics

2.2.5 Distribution by Programming Languages

The ProgCont system can be used to submit source code for solving tasks in 7 different programming languages. These were divided into 5 categories:

- C: ANSI C (C89) and C (C99)
- CPP: C++,
- JAVA + CS: C# and Java,
- Pascal,
- Python.

Figure 5 shows the percentage distribution of submissions by language, and we have grouped submissions in Java and C# into one category (since IT also typically refers to these languages as one). Because there have been very few solutions in Python, we will not consider them in subsequent analyses. One reason for this is that Python language support did not exist until October 2016, while at least one programming language in the other 4 categories was available throughout the period under consideration.

74% of the submitted solutions were in C language, which correlates with the distribution of the programming languages taught in the University, because the subjects *Introduction to Programming* and *Programming Languages 1* focus on C language, and so the students practice more in this language.

This is especially important to keep in mind when examining the distribution of the preferred programming languages for math problems. Typically, the C language is not what we would recommend for solving mathematical tasks. Yet, it is strongly over-represented simply because our students only know this when they encounter these tasks.

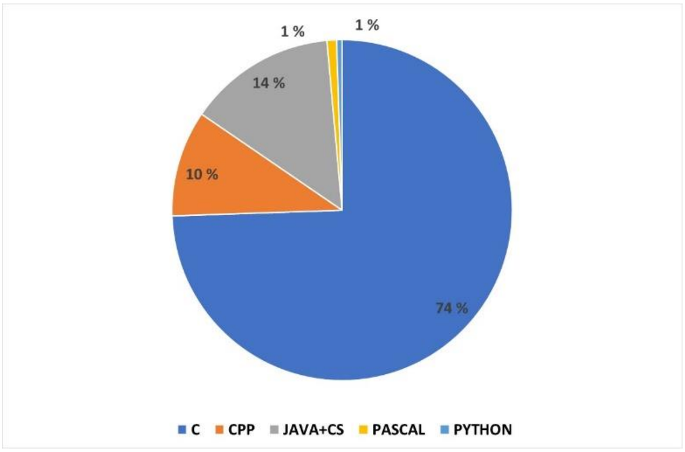

Figure 8
Distribution of submissions in different programming language

Figure 9 shows the distribution of programming languages in each of the two categories. The data show that our Hypothesis 3, according to which there is a programming language specific to the disciplinary field has been confirmed: C and Pascal are more common for mathematical tasks, and an object-oriented language (C++, Java, or C#) is more common for IT tasks (Figure 9 and Figure 10). This supports a recent decision to make students learn Python as their first language.

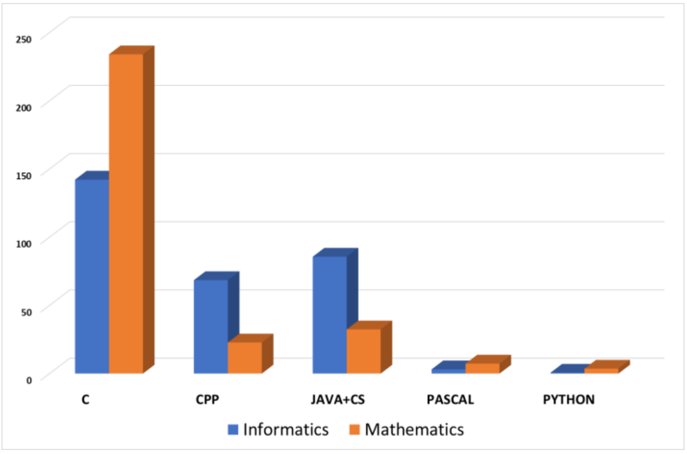

Figure 9
Distribution by programming languages in the two disciplines

The subject of our present study is only those tasks that could be solved in any programming language. Among our students, those who have already mastered an object-oriented language should also be able to program in at least one structured programming language. In their case, the choice of programming language may be more conscious, however, many students have dealt with these tasks who are not yet familiar with an object-oriented language, so it is not worth concluding the exceptionally high number of submissions in C. (The ratio of submission numbers is only interesting in comparing the two types of tasks.)

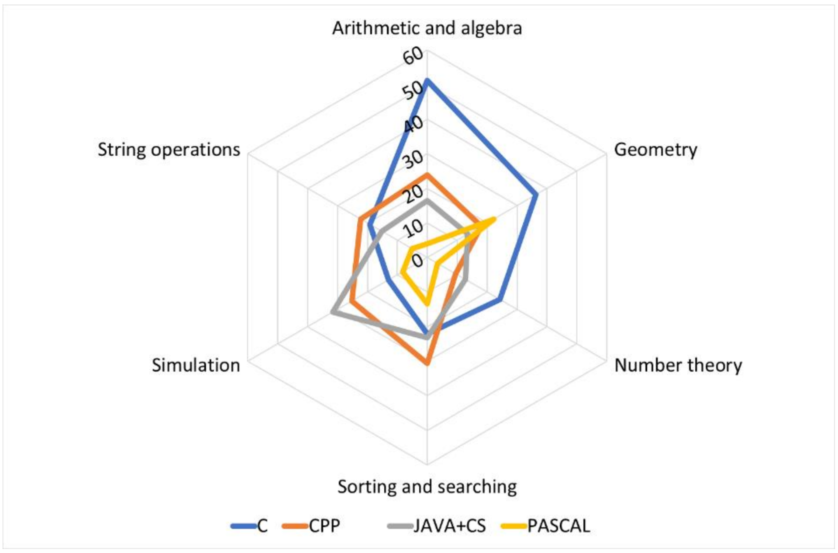

Figure 10

Distribution of submitted solutions by programming languages and task categories

In the left part of Figure 11, we compare the ratio of submitted solutions (outer ring) and accepted submissions (inner ring) for mathematical topics and do the same on the right for IT topics. It is clear that in both cases the C programming language leads in terms of the number of submissions. It is quite striking that in the case of tasks that also require mathematical skills, the acceptance rate of solutions in C programming language is remarkably high: 0.3689 (912 out of 2 472). In this respect, the IT tasks solved in C++ programming language are in second place with only 0.25 (107 out of 428).

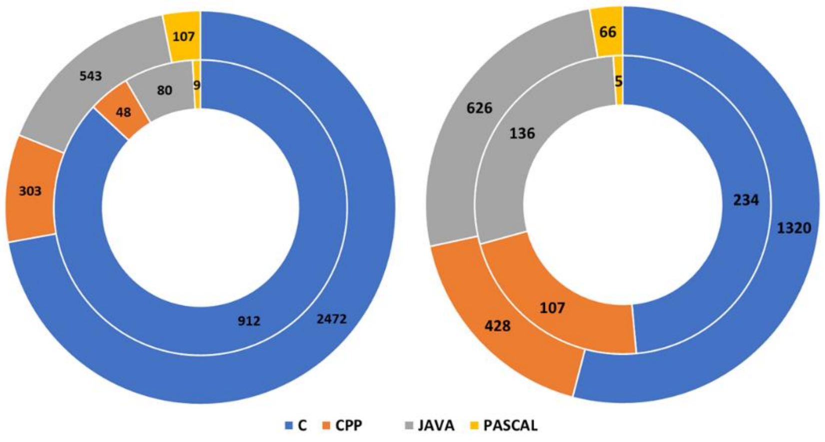

Figure 11
The ratio of total and successful submissions by programming language

Conclusions

In our article, we performed an analysis of a well-defined group of automatically evaluated programming tasks using the ProgCont system of the University of Debrecen, looking for the relationship between the performance of tasks requiring mathematical and IT knowledge. From the more than 277 000 automatically evaluated submitted solutions, we have selected the nearly 6 000 that belonged to the properly categorized tasks. We had previously set up 3 hypotheses, which were tested.

We found that, contrary to our expectations, users perform better on tasks that require proficiency in both disciplines. Therefore, it seems worthwhile to place more emphasis on tasks that require math skills in the teaching of programming, which also develops students' mathability skills [1]-[10], [18]. At the same time, it is worth rethinking the choice of the first programming language taught (as is currently the case in some of our majors).

As a result of our research, we have identified those task types (within the examined task types) where the performance of the users is weaker. At the same time, we set the direction for expanding the set of tasks that form the basis of the research.

We identified our users' preferences for the choice of programming language for the two large categories of the examined tasks. We have seen that the choice is not the best right now. This needs to be corrected during education.

Acknowledgement

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

References

[1] Baranyi, P., Csapó A.: Definition and Synergies of Cognitive Infocommunications, Acta Polytechnica Hungarica, 2012, 9, 67-83

[2] Baranyi, P., Gilanyi, A.: Mathability: Emulating and Enhancing Human Mathematical Capabilities, 4th IEEE International Conference on Cognitive Infocommunications, 2013, 555-558

[3] Baranyi, P., Csapó, A., Várlaki, P.: An Overview of Research Trends in CogInfoCom. In: Szakál A (ed.) 18th International Conference on Intelligent Engineering Systems - INES 2014, Tihany: IEEE Hungary Section, 2014, pp. 181-186

[4] Biró, P., Csernoch, M., Abari, K., Máth, J.: First Year Students' Algorithmic Skills in Tertiary Computer Science Education. In: Kunifuji, S., Papadopoulos, G., Skulimowski, A., Kacprzyk, J. (eds) Knowledge, Information and Creativity Support Systems. Advances in Intelligent Systems and Computing, Vol. 416, Springer, Cham., 2016, https://doi.org/10.1007/978-3-319-27478-2_24

[5] Biró, P., Csernoch, M.: The mathability of computer problem solving approaches, In 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom2015) (pp. 111-114), http://doi.org/10.1109/CogInfoCom.2015.7390574

[6] Biró, P., Csernoch, M.: The mathability of spreadsheet tools, In 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom2015) (pp. 105-110), http://doi.org/10.1109/CogInfoCom.2015.7390573

[7] Chmielewska, K., Gilányi, A., Łukasiewicz, A.: Mathability and Mathematical Cognition, in 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2016, 245-250

[8] Chmielewska, K., Gilányi, A.: Educational context of mathability. Acta Polytechnica Hungarica 15 (5), 223-237

[9] Chmielewska, K., Gilányi, A.: Mathability and Computer-aided Mathematical Education, in 6th IEEE Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2015, 473-477

[10] Chmielewska, K., Matuszak, D.: Mathability and Coaching, in 8th IEEE Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2017, 427-431

[11] Falus, I.: Introduction to the methodology of pedagogical research. (Bevezetés a pedagógiai kutatás módszereibe) Műszaki Kiadó, Budapest, 2004

[12] Futschek, G.: Algorithmic Thinking: The Key for Understanding Computer Science. Informatics Education–The Bridge between Using and Understanding Computers, 4226, 2006, 159-168

[13] Kádek, T., Biró, P.: On the way to the study group application with the ProgCont API. (Úton a szakköralkalmazás felé a ProgCont API-val.) INFODIDAKT 2019 konferencia, Zamárdi, Webdidaktika Alapítvány, https://people.inf.elte.hu/szlavi/InfoDidact19/Manuscripts/KTBP.pdf

[14] Kádek, T., Biró, P.: The ProgCont API: innovative evaluation of solutions to programming assignments. (A ProgCont API: programozási feladatok megoldásainak újszerű kiértékelése.) SZÁMOKT 2019, Temesvár, Románia: Erdélyi Magyar Műszaki Tudományos Társaság (EMT) kiadó, 2019, 191-195

[15] Kádek, T., Biró, P.: Effects of distance education on the ProgCont system. (A távolléti oktatás hatásai a ProgCont rendszerre.) In: ENELKO SzámOkt 2020, Erdélyi Magyar Műszaki Tudományos Társaság EMT, Kolozsvár, 2020, 104-109

[16] Kádek, T., Kósa, M., Pánovics, J.: Informatics competition tasks. (Informatikai versenyfeladatok.) Gyires Béla Informatikai Tananyag Tárház, 2014, https://gyires.inf.unideb.hu/GyBITT/04/

[17] Kósa, M., Pánovics, J., Gunda, L.: An Evaluating Tool for Programming Contests. Teaching Mathematics and Computer Science, 3/1, 2005, 103-119

[18] Kővári, A., Rajcsányi-Molnár, M.: Mathability and Creative Problem Solving in the MaTech Math Competition. Acta Polytechnica Hungarica, 2020, 17, 147-161

[19] Pólya, G.: How to solve it, United States of America: Princeton University Press, 1957

[20] Wing, J. M.: Computational thinking. Communications of the ACM, 49(3), 2006, 33-35, https://doi.org/10.1145/1118178.1118215