Bringing Algorithms to Flemish Classrooms: Teaching the Teachers, and some Students

Bern Martens, Bart Demoen, Febe Karpez, Dorien Vandenhove, Kristien Van Loon
KU Leuven, Department of Computer Science, 3001 Heverlee, Belgium

ABSTRACT

Computer science (CS) is currently not yet part of the official curriculum imposed by the Flemish government for secondary education. However, an increasing number of schools offer the topic, or elements of it, in a "free" course, especially in grades 7 and 8, and also in scientific and/or technical profiles in grades 9 to 12. The teachers who develop and teach these courses usually do so with great effort and enthusiasm, but a limited background in CS. The universities of Leuven, Gent and Hasselt, are organising several series of workshops in which teachers' skills and knowledge on Physical Computing and Algorithms are enhanced during in-service training. The approach taken is presented, as well as the results of a small scale teaching experiment on algorithmics in grade 12, and plans for future activities and research in this area.

CCS CONCEPTS

• **Social and professional topics → Computing education**; **Model curricula**; *Computational thinking*; *Computing education programs*;
• **Theory of computation** → *Design and analysis of algorithms*;

KEYWORDS

Computing education, Algorithms, In-service teacher training

ACM Reference Format:
Bern Martens, Bart Demoen, Febe Karpez, Dorien Vandenhove, Kristien Van Loon. 2017. Bringing Algorithms to Flemish Classrooms: Teaching the Teachers, and some Students. In *Proceedings of WiPSCE '17, Nijmegen, Netherlands, November 8–10, 2017,* 2 pages.
https://doi.org/10.1145/3137065.3137070

1 INTRODUCTION

In many countries throughout the world computational thinking, computer science and/or programming are high on the agenda of curricular reform in K-12 education.

In Belgium, education is organised by the three language communities separately: Dutch, French, German. The largest of them is the Dutch language community, also called "Flemish community". In Flemish official education programmes currently only some specialised vocational study profiles in grades 11 and 12 focus on computer science [4]. Still, many individual teachers, teacher teams, schools and even school networks are currently introducing classes and courses on computer science topics without official curricular guidelines. In some cases, efforts are research based [6], but mostly, teachers improvise and often lack much schooling in computer science, let alone its teaching methodology.

The Departments of Computer Science and Computer Science Teacher Education of the Universities of Leuven, Gent and Hasselt have joined forces, with the support of Google's CS4HS programme, to provide in-service training to teachers interested in developing computer science classes and courses, as separate subjects and/or as crucial ingredients of integrated STEM (iSTEM) education.

In Section 2, we present the resulting Progra-MEER initiative. We briefly report on its overall aims and activities, its results so far and some plans for the near future. We explore in more detail a strand on algorithms that we newly implemented in our training programme from January to May 2017. Next, in Section 3, we report on a specific, limited but nevertheless interesting, experiment with teaching algorithmics in grade 12 by two (pre-service) students in the Leuven CS teacher education.

2 TEACHING THE TEACHERS: THE PROGRA-MEER INITIATIVE

In 2015, we founded the consortium Progra-MEER, and organised a series of 5 workshops on *Physical Computing* (PC - see for instance [5]) for teachers, in Gent as well as in Leuven. PC is relevant for CS, as well as for iSTEM. PC is attractive to many students, but teachers really need to learn about the underlying concepts, methods and teaching materials. As our longer-term goal is a decent Flemish school curriculum in CS, we felt that the PC workshops should be complemented with workshops on Algorithmics. We did so in school year 2016-2017, in three locations (Hasselt joined).

The goal of the Algorithmics workshops was two-fold: (1) to enhance the participants' notion of algorithms, to learn about the properties and limitations of algorithms, to acquire skills in reading, programming and analysing algorithms, and to get acquainted with some classical algorithms and their applications; (2) to develop a classroom project for their own students, share it with each other, and present it in a joint final workshop.

Each workshop series consisted of 4 afternoon sessions. A recurring activity in each session went as follows. A particular (instance of a) problem or puzzle was presented to the participants. They were expected to solve it, describe their solution method, formulate it as an algorithm, reason about whether it always works, etc. This activity led to the notions of abstraction, problem representation, correctness and efficiency. Interleaved with this, the workshop leader discussed these notions explicitly, and by asking questions led the participants to the discovery of new concepts, techniques, generalisations and real life applications.

The workshop leader explained just enough concepts to lead up to the P versus NP question. Some time was spent on correctness and termination, and for two algorithms, these properties were

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WiPSCE '17, November 8–10, 2017, Nijmegen, Netherlands
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5428-8/17/11.
https://doi.org/10.1145/3137065.3137070

worked out completely, and through the beautiful contradiction app formerly described on thorehusfeldt.net/2012/06/25/the-freeze-app-does-not-exist, the participants discovered that some problems have no algorithmic solution.

Throughout, the participants were encouraged to refine and discuss their project. Inspiration for the projects was drawn from different sources amongst which the *Python Mode for Processing* (processing.org) and a book on the importance of algorithms [3].

The background of the Leuven participants, as well as their teaching context, was very diverse. No-one was acquainted with systematic approaches to algorithms and programming (in the small), neither with computation theory. The participants described the series of workshops as *challenging, but doable*. They also indicated that, to their surprise, coding is not the first part of solving a problem by computer, but that the analysis of the problem, its abstraction and the design of an algorithm comes first, and that these activities can be done *on paper*. The sequence of an unplugged activity and a guided group discussion led to the formulation of an algorithm and sometimes even to a Python program, and involved the participants to a degree that kept them motivated.

3 STUDENTS AND THE STABLE MARRIAGE ALGORITHM: A HAPPY MARRIAGE?!

To test our ideas about classes on algorithmics in Flemish secondary education, we developed a two hour class on the stable marriage problem and the Gale and Shapley algorithm to solve it [1, 2] in all 17 students with a mathematically strong profile in 12th grade were obliged to participate. These 17 students previously had about 10 hours of class on programming in grade 10. Six of them, with a strong interest in computer science, had gained a richer experience with programming on their own initiative.

Before our intervention, we asked the students to fill out a questionnaire on a number of issues connected to mathematics and computer science. There was wide agreement on the relevance of both mathematics and computer science, and the two subjects were perceived to have a lot in common, but opinions were divided on whether it was fun to study them. Finally, more than 2/3 of the students wanted "informatics" to focus less on the use of applications.

The intervention comprised two classes of 50 minutes each with a short break in between. In the first class, students were introduced to the stable marriage problem and the Gale and Shapley algorithm through an unplugged coupling activity “as they saw fit” followed by an unplugged execution of the algorithm and a simulation done individually by each student with pencil and paper. In the second class, after the short break, the students sat in four groups and were challenged to prove, through logical reasoning, that the algorithm always stops, always produces a stable outcome, and performs at most n^2-2n+2 cycles, where n is the number of "men" and "women". Finally, the teachers touched upon the optimality issue, observing together with the students that those who propose seem to be the ones who get their best possible choice, but not formally proving this, and signalling some practical applications as well as possible extensions and variants.

One to two weeks after the intervention, 14 students filled out a second questionnaire. All phases of both classes were enjoyed by 11 of those 14 students. And no less than 13 students reported having learned significantly during especially the second class, on proving the properties of the algorithm, and discovering its applications and extensions. Even though the students had a mathematically strong profile, they considered the proving part to be the hardest of the whole experience: 6 students rated it as “hard”, 7 as “just hard enough”, and only one as “easy”.

These findings were complemented through live discussions in small groups. Especially students with a pronounced interest in computer science and a more extensive prior knowledge of programming suggested speeding up the first part; the others valued the "gentle" introduction it provided. A spectacular finding lies in the fact that all students reported constructing "mathematical proofs" through logic reasoning as a completely novel, and challenging, but interesting experience! Finally, 12 of the 14 students would like to see more classes like this. To the best of our knowledge, the above described experimental class was the first in Flemish secondary education on a "classical" algorithm and the formal proof of some of its properties. Due to its small scale, our experiment precludes grand conclusions, but we feel strengthened in our resolve to push for more and better education in this area.

4 CONCLUSION AND FURTHER WORK

In this poster paper, we presented a brief sketch of the Progra-MEER initiative offering in-service training to (future) teachers of computer science and programming in Flemish secondary education. We complemented this with a somewhat more extensive report on one particular experiment with a class on algorithmics by two students in teacher education.

In the near future, we hope not only to continue but also expand our efforts along several lines. First, we want to give support to building a long term community of Flemish CS teachers, and started doing that through our cooperation with 2Link2 (2link2.be), our annual Progra-MEER final events where participants in the in-service training present their work and projects to each other, the Progra-MEER organisers and a larger audience, and the program-uurtje.org website where developed materials are made available for other teachers to use in their classrooms. Secondly, we want to start a specific thread of training for teachers working with students in grades 5 tot 8. This is part of a grander aim of contributing to the further development of a well thought through CS curriculum, fit for the Flemish context, from Kindergarten to grade 12. And last but not least, we hope to further develop our fledgling research on computer science teaching materials and methodology.

REFERENCES

[1] C. T. Bergstrom, T. C. Bergstrom, and R. J. Garratt. 2013. Choosing Partners: A Classroom Experiment. *The Journal of Economic Education* 44(1) (2013), 47–57.
[2] D. Gale and L. S. Shapley. 1962. College Admissions and the Stability of Marriage. *American Mathematical Monthly.* 69 (1962), 9–14.
[3] J. MacCormick. 2012. *9 Algorithms that Changed the Future.* Princeton Uni. Press.
[4] Bern Martens and Tom Hofkens. 2013. Positioning Computer Science in Flemish K-12 Education: a Reflection. In *Proceedings of WIPSCE 2013.* 35–36.
[5] Mareen Przybylla and Ralf Romeike. 2014. Physical Computing in Computer Science Education. In *Proceedings of WiPSCE 2014.* 136–137.
[6] Francis wyffels, Bern Martens, and Stefan Lemmens. 2014. Starting from Scratch: Experimenting with Computer Science in Flemish Secondary Education. In *Proceedings of WIPSCE 2014.* 12–15.