New Methods for Teaching Data Structures and Algorithms

Lauri Malmi, Ari Korhonen
Department of Computer Science and Engineering
Helsinki University of Technology
Finland
{lma,archie}@cs.hut.fi

Abstract

In this paper, an implementation of a data structures and algorithms course is presented. The course is based on the constructivistic learning paradigm. The students survey the contents of the course by simulating algorithms, by designing algorithmic solutions for practical applications, and by carrying out a small research project on the efficiency of algorithms. An internet-based learning environment is an integral part of the course, because it provides an opportunity to apply activating learning methods on a mass course with hundreds of students.

1 Introduction

Data structures and algorithms are core issues in computer science. Knowledge about their properties, efficiency and applicabability to different practical cases is needed in all software projects. Therefore, a course of this topic is given as the second computer science course in Helsinki University of Technology (HUT), after the students have learned the basic programming skills.

The goals of the course are the following. First, the students should learn the working of basic data structures and algorithms in different topics, like sorting, searching and graph applications. Second, they must understand the principles how the performance of different algorithms is evaluated analytically, and - what is important - how the limitations of algorithm analysis affects the interpretation of analytical results. Third, students of computer science curriculum should understand the principles and problems of experimental algorithm analysis. Knowledge about analysis methods is important, since the performance of different algorithms solving exactly the same problem can vary orders of several magnitudes.

About 600 students are enrolled on the course yearly. One third of them study in computer science curriculum and the rest study in other curricula in HUT. Therefore the course is split into parts. The general course, data structures and algorithms, called DSA, (3 credits) is directed to all students, and the following laboratory course (2-5 credits) is directed for computer science students only.

We have developed solutions which aid us to activate the students and to provide feedback for them even though the number of students in DSA is large. First, telematic tools are heavily used in communication with and between the students [3]. Second, we have designed and implemented dedicated software for giving feedback for the students [1, 2, 4]. Third, several activating working methods are used in both courses. As the results of these actions we have observed very good learning results and positive feedback from the students.

2 Didactic Overview

When designing the course we have considered the following issues very important. First, the students should be active during the whole course in exercising new issues they have learned. Second, they must apply their knowledge to design solutions to non-trivial problems. Third, they must analyze their design and state their arguments about its pros and cons. Fourth, they must actively do research on some spesific topic covered in the course. Finally, feedback and guidance is given in all phases of the course.

Group work is considered important, and peer reviewing is used to help the generation of new ideas for the students and to support criticism among them.

This didactic design of the course has not been established by initially applying a certain learning theory or paradigm. It is the result of an active development of the course during the whole 90's. In its current form it, however, has many connections with constructivistic learning theory [5].

3 Working Methods

The DSA course has the following working methods. Lectures are used to give an overview of the contents of the course and to explain important and difficult topics. Many problems are discussed during the lectures to raise the motivation and attention of the students. Moreover, after the teacher has explained some important topic, the students are given a simple problem which they have to solve using the new information. After this the teacher explains the answer and discusses it with the students, if necessary. The students have access to a printed lecture summary prepared by the teacher which they can fulfill on the lectures. The text book is used as a supplementary material which covers the details more thoroughly than in the lecture summary.

In the basic exercises the students simulate important algorithms covered in the course. They apply given algorithms by showing how they change given data structures. To be able to solve these exercises they have to understand the key points of the algorithms, although they are not required to implement them in code.

It is essential that they receive feedback of their solutions. This is accomplished by using the Trakla system [1, 2] which fully computerizes these exercises. The exercises are published in the WWW pages of the course and the students can solve them either by using a graphical user interface called Trakla-Edit [2] or by sending their solutions directly to Trakla server by email. In both cases, the server immediately evaluates the solutions by comparing them to model solutions and sends feedback how well these matched each other. In case the answer was erroneous, the students may refine their solution a few times. It is notable that no trial-and-error method is feasible in this refinement process since the solution space is huge. The students must rethink their answer, which supports learning. Note that the cycle submit - feedback - resubmit would not be feasible with any practical human resources due to the time delay in giving the feedback.

Another advantage of the computerization of the exercises is that each assignment is tailored individually for each student, i.e., they solve the same assignment with different initial data. No copying of answers is possible. Instead, natural co-operation is encouraged unless a student fully solves the assignments of another student.

The next phase in the course is the design problem, where the students have to design in groups of 2-3 persons a solution (though not implement it) for a realistic non-trivial problem. Here they have to apply their knowledge about basic algorithms by combining them or by modifying them to solve the new problem. In addition, they have to analyze their solution and state its pros and cons. The problems have been set up so that no obvious solution is available. Instead, the students have to identify possible tradeoffs caused by contradictory requirements.

An essential part of the work of a designer is to present ones ideas in a clear and understandable form. To support this the student groups review the works of a few other groups. Technically this is arranged as follows. Each group submits their design to us as an HTML page (URL) accessible in the Web. We copy the pages to our server, anonymize them, give them a new numeric URL address and send a few such addresses to each group by email. Each group then reviews these works, prepares a written evalution of each work, proposes the best one and submits their evaluation report to us as an HTML page. These reports are then sent to the original design groups. We evaluate both the original design papers and the review reports.

This working method has the following advantages. First, the students understand the problem better when they see different solutions to it. Second, they realize the difficulty of expressing one's thoughts clearly and how important it is. Third, they have to investigate whether a proposed solution works or not, i.e., they have to exercise criticism and evaluation, which is not trivial in this case.

The course has also an examination, where most assignments require a synthesis of some parts of the course contents. In 1999 an option was provided to replace the exam by preparing a portfolio of the course.

In the laboratory course the student groups carry out a small-scale experimental research project. First, they survey a number of algorithms solving the same problem, e.g., a number of sorting algorithms, and discuss their analytical properties. Second, they prepare a test plan where they discuss the problem of comparing the performance of these algorithms experimentally, identify the possible parameters for test input data, choose the factors to be considered in the tests and propose their values. This test plan is evaluated and critisized by the teacher after which the students refine the test plan and perform the tests. Finally they write an article of their research in a scientific style. The articles are reviewed by the course staff and the best of them are presented in a small conference and published as an on-line proceedings in the web.

From a didactic point of view the laboratory course has three aims. First, the students learn more thoroughly a number of algorithms when they really implement them. Second, they encounter the difficulty and pits of experimental algorithm analysis under the proper guidance of the teacher. Finally, and which is actually the most important issue, they exercise scientific working and writing, already when they are studying their second or third year.

4 Discussion

The course has been organized so that the learning process of the students proceeds in phases. Lectures provide the motivation and the orientation basis for the course contents. Key topics are also discussed and exercised in them. In the basic exercises the students have to apply information of all basic algorithms more thoroughly. After learning them they can apply them as tools for solving the more advanced design problem. In this phase they also have to exercise analysis and argumentation which is highly important for a good design. In the laboratory course this process is continued with a true exercise in scientific working. Thus, as in the constructivistic learning paradigm, we consider essential that the students are active in their learning process. The key role of the teacher is to set up the assignments and to provide guidance and feedback for the students.

For the basic exercises it is easy to measure the learning results. In years 1998 and 1999 each student solved about 25 different assignments. More than half of them got at least 90 percent of maximum points and about 2/3 of them got at least 80 percent of maximum. This effectively guaranteed that the basic knowledge of how algorithms work had been adopted very well.

For the more advanced phases of the course, such a numerical measurement of results is not feasible. Almost all student groups presented a working solution for the design problem, but the presentation level varied a lot. The review process was too demanding for some groups, but it is not clear how much of this results from the poor presentation of the reviewed design. In the laboratory course the level of the final article varied quite much, as well. However, all groups presented acceptable work and about one third of the papers were well written. The best ones could be candidates for a true conference submission.

The course is constantly evolving. At its current status the assignments have mostly been given by the teacher. In the future, the students will have more freedom in choosing them. The internet-based learning environment Trakla is developed towards a more versatile self-contained environment where the students may exercise various topics on their own [4].

References

[1] Hyvönen J., Malmi L.: *TRAKLA - A System for Teaching Algorithms Using Email and a Graphical Editor*. Proceedings of HYPERMEDIA in Vaasa'93, pp. 141–147, 1993.

[2] Korhonen A.: *World Wide Web in Computer-Aided Learning of Algorithms and Data Structures*. MSc Thesis, Department of Computer Science, Helsinki University of Technology, Finland (in finnish), 1997.

[3] Korhonen A., Malmi L., Internet-Based Learning of Data Structures and Algorithms. To appear in a publication of selected papers from Conference on Learning and Collaboratorion in Virtual Environments, University of Jyväskylä, Finland, 1999.

[4] The LEAD project home page. See, http://www.cs.hut.fi/ tred/LEAD/

[5] Rauste-von Wright M., von Wright J. Oppiminen ja koulutus, WSOY, Finland, 1998.