World Transactions on Engineering and Technology Education *© 2014 WIETE*
Vol.12, No.3, 2014

Research-oriented teaching of an *Algorithm Design and Analysis* course

Xuemin Zhang, Zenggang Xiong, Yaomin Ding, Guangwei Wang & Conghuan Ye

Hubei Engineering University
Xiaogan City, People's Republic of China

ABSTRACT: *Algorithm Design and Analysis* is a course characterised by an integration of application with creativity and with a requirement that students have a good foundation in mathematics. An analysis of the problems encountered in the teaching of algorithm design and analysis is presented in this article, as is a summary of the teaching content, methods and means. Then, the importance of introducing research-oriented teaching is explained. Finally, the research-oriented teaching reforms of the course are proposed.

INTRODUCTION

Algorithm Design and Analysis is a course with closely related theory and practice and is one of the core courses in computer science and technology. Algorithm design and analysis cultivates computer science and software engineering talent for the new century. It is necessary for both application-oriented and theory students to understand and master the basic methods of algorithm design and algorithm complexity. Through the Algorithm Design and Analysis course, students master common algorithms in computer software design and can analyse the complexity of the algorithms to help optimise solutions in terms of speed and cost for practical problems.

Algorithm Design and Analysis is an application-oriented course at the core of computer science and technology that includes computer algorithms, algorithm design and the computational complexity of algorithms. This lays a solid foundation for independently designing algorithms and analysing their complexity. This is essential for scientific and technical work involving computer systems architecture, system software, and application software research and development [1]. There is a rich source of algorithms, given their continued study in China and abroad. It is imperative that the course content and methods for teaching algorithms enable students to not only master the basic knowledge, but also the frontiers of knowledge. Drawing on actual teaching experience, presented in this article is a preliminary study on algorithm analysis and design in undergraduate teaching in terms of teaching content, teaching methods, algorithm practices and assessment methods.

PROBLEMS IN TEACHING ALGORITHM DESIGN AND ANALYSIS

The course has many knowledge points and a sound theoretical basis. The course content of algorithm design and analysis includes classic algorithm design techniques, such as recursion, and divide and conquer, dynamic programming, greedy algorithms, backtracking, branch and bound, and graph algorithms. It also includes a number of advanced algorithm designs, such as network flow and matching, heuristic search, and linear programming.

The course requires a highly abstract and logical thinking ability, which makes it difficult to learn. The credit hours for the Algorithm Design and Analysis course are limited at Hubei Engineering University. Time is very short for students in which to master the variety of algorithm design and analysis methods.

High Requirements for Practice, Lack of Innovation

Algorithm Design and Analysis is a computer course, with a high practical content. Students need problem-solving and programming language skills. However, students' knowledge varies and they lack the ability to think independently. Hence, many students cannot complete the course practical work.

Unidirectional Teaching and Poor Learning Motivation

Teaching, in which teachers talk at students who listen passively, leads to students with an apparent lack of learning initiative. Due to a lack of communication between students and teachers, students passively learn the course, which is not conducive for algorithm design and analysis learning.

THE NECESSITY FOR INTRODUCING RESEARCH-ORIENTED TEACHING OF THE ALGORITHM DESIGN AND ANALYSIS COURSE

The Place of Algorithm Design and Analysis in the Computer Curriculum

Algorithm Design and Analysis is a core course in the computing curriculum, as shown in Figure 1. The prerequisite courses include discrete mathematics, programming, computational methods and data structures. The follow-on courses include compiler theory, operating systems, software engineering and data mining. Learning algorithms directly affects the learning of the follow-on courses. Essentially, algorithms are an all-purpose tool, which students need to fully grasp to become experts in other specific areas. The algorithm's course is at the core of the curriculum system and so is very important for computer students [2].

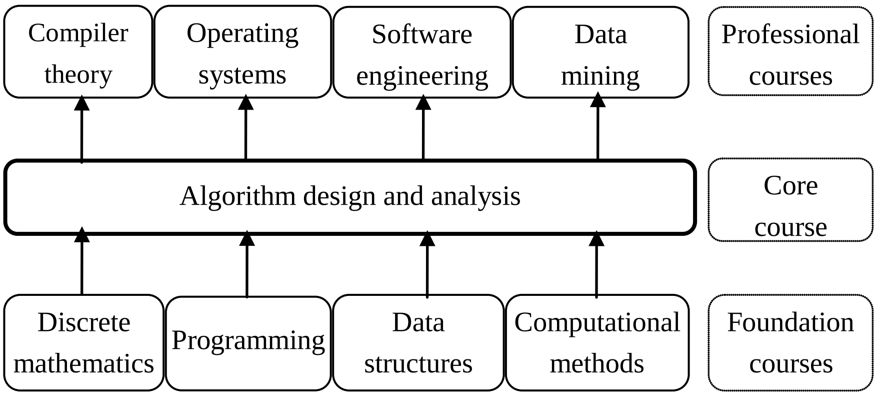

Figure 1: Algorithm design and analysis in the computer curriculum.

Concepts of Research-oriented Teaching

Research-oriented teaching combines teaching with learning. In research-oriented teaching, teachers assume students have basic knowledge and use problems to stimulate their interest and enthusiasm. They, then, guide students to take the initiative on exploring and solving the problems. The objective of research-oriented teaching is to train students in innovative research-like thinking. This requires students to acquire new information and incorporate it into their *corpus* of knowledge. The basic approach is to guide students in exploring and experiencing the process of knowledge acquisition through the process of identifying questions, and analysing and solving problems. Hence students can build on the knowledge of predecessors.

The Necessity for Introducing Research-oriented Teaching in Algorithm Design and Analysis

Algorithm Design and Analysis is a design-oriented core course that is closely related to many disciplines, such as information science, computing, management, business, engineering and technology. The course has the following characteristics:

- Both theory and practice are required. Solving practical problems is the goal, so that the study of algorithm analysis and design, and practical problem-solving are closely related.
- The need to master mathematical knowledge to understand classical algorithms and their complexity.
- The course is creative.

The main teaching process of the Algorithm Design and Analysis course is summarised in Figure 2.

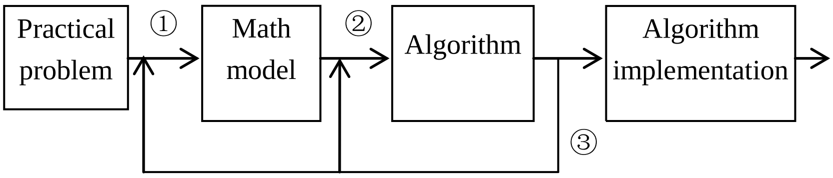

Figure 2: The teaching process of the Algorithm Design and Analysis course.

In the first step, teachers guide students toward analysing a practical problem and producing a mathematical model of the problem. In the second step, teachers guide students toward using a variety of algorithm design techniques and strategies to complete the algorithm design based on the mathematical model. In the third step, the teacher analyses the algorithm that has been designed. Algorithm analysis here refers to pre-analysis, against the abstract model. Algorithm analysis is important in raising questions, identifying problems and stimulating creative thinking. This process may take the student back to the second step in the search for a more efficient algorithm or even to the first step, to establish a new mathematical model.

Thus, the introduction of a research-oriented teaching mode for algorithm design and analysis should improve research-thinking and innovative ability.

SELECTION OF TEACHING CONTENT FOR THE ALGORITHM DESIGN AND ANALYSIS COURSE

Algorithm Design and Analysis is one of the more difficult computer science courses, since it requires students to have good mathematical knowledge, abstract thinking abilities and programming skills. Many students with interest in the course are prone to weariness as the material increases in difficulty, and they may even give up studying the course. Carefully selected teaching content is required in order to alleviate these problems [3].

Selecting Classical Algorithms

As previously mentioned, algorithm design and analysis requires good logical thinking ability, mathematical modeling and analysis capabilities, as well as programming skills. Developing the content of this course requires the analysis of classical algorithms, such as divide and conquer, dynamic programming, greedy algorithms and backtracking, for potential use in the course.

During the course, one knowledge point can be associated with related knowledge points, e.g. solving the knapsack problem with dynamic programming, to make students think about how to use the greedy algorithm and backtracking to solve the problem. By comparing several different strategies to solve the same problem using different methods, students come to better understand the different algorithms, as well as improving their abilities to analyse and solve problems.

Developing Specific Application Scenarios to Explain the Algorithms

The teaching of algorithm design and analysis should include practical application scenarios, in order to stimulate student interest and to change passive into active learning. This will facilitate a student’s understanding of the algorithm. The most important five kinds of design strategy for algorithms (i.e. iterative, brute force, divide and conquer, greedy method and dynamic programming) is still the focus of the teaching, but they should be explained with examples to increase students’ interest. For example, explain divide and conquer to design a sort algorithm, by citing the PageRank algorithm of the Google search engine as a real-life example.

Introducing the Latest from Science and Technology

In recent years, the focus of the study of algorithms has included randomised algorithms, approximation computing, parallel computing and artificial intelligence. These algorithms are more likely to resonate with students and arouse their interest. Therefore, apart from explaining the classical algorithms, teachers may selectively introduce some of the latest research from academia and industry. This will make students feel they are not far from the forefront of the discipline. When explaining these algorithms, teachers can encourage students to think creatively, and teachers can possibly identify innovative students.

Dealing with the Relationship between Algorithm Design, Analysis and Data Structures

Before learning algorithm design and analysis, students should finish learning about data structures. These two courses overlap, such as traversals and shortest path, so when selecting material for the Algorithm Design and Analysis course consideration needs to be given not only for the students to be able to grasp the ideas of algorithm design, analysis and algorithm performance, but also to avoid students feeling that material is being repeated. This requires dealing well with the relationship between the two courses when setting the syllabi.

In a data structure course, the definition and representation of a static data structure is paramount, while the algorithms are supplementary. In an algorithm course, students design algorithms based on a specific data structure. The teaching focuses on guiding students toward analysing and comparing the performance of various algorithms.

Designing Practical Projects for Algorithm Design and Analysis

The practical teaching of algorithm design and analysis follows the principle of individualised teaching. Experiments are set to explore the important concepts and methods of algorithm design and analysis. These include verification experiments, design experiments and comprehensive experiments, from the divide and conquer strategy, greedy algorithms, dynamic programming and backtracking, and branch and bound. These guide students toward exploring the design principles and analysis techniques of algorithms, in order to deepen their understanding of the concepts, methods of algorithm to improve students' analysis, design capabilities of software systems and practical research ability. These projects are flexible and scalable, so that teachers can dynamically adjust projects based on students' abilities. Experimental projects of the Algorithm Design and Analysis course are shown in Table 1.

Table 1: Experiment projects of the Algorithm Design and Analysis course.

Number	Experiment name	Experiment request	Experiment type
1	Implementation of merge sort algorithm	Using recursive and non-recursive bottom-up processes, students should master the process of recursive algorithms	Verification
2	Realise the quick sort with divide and conquer	Students should understand the quick sort algorithm and master how to solve problems using algorithm design	Verification
3	Realise dynamic programming method of the longest common subsequence	Students should master the basic strategy of the design of a dynamic programming algorithm	Verification
4	Park Travel Guide: provide interactively a tourist's walk for the shortest path within shortest time from the tourist's location to the destination	Students should master the basic design idea of greedy algorithms and apply it to solving practical problems	Verification and design
5	Huffman coding algorithm of trigeminal tree; make time, spatial analyses	Students should master the basic idea of Huffman coding and the greedy algorithm design	Verification and design
6	Realise M graph colouring problem with backtracking	Students should master the basic idea and framework of the backtracking algorithm	Comprehensive
7	Solve the shortest path problem with a branch and bound algorithm	Students should master the basic idea of branch and bound algorithms	Verification

REFORM OF THE TEACHING OF THE ALGORITHM DESIGN AND ANALYSIS COURSE

Using Innovative Teaching Methods in a Learner-centred Approach

In traditional classroom teaching, teachers teach and students learn. Teachers select teaching materials, organise teaching and, then, pass knowledge on to students. Students largely passively accept knowledge without forming their own ideas. But the important teaching objective of the Algorithm Design and Analysis course is to cultivate students' ability to analysis and design algorithms, which runs counter to this teaching model.

In the field of algorithms, students are a *tabula rasa,* who think about the problems without constraint. A young, active thinker may come up with ideas that even teachers will never have thought of. Therefore, in the classroom, teachers should change roles, from knowledge imparters to knowledge guides. The teacher's task is to introduce problems, and to provide help through discussion and co-operation, so as to mobilise the student's initiative.

Promoting Research-oriented Teaching

In research-oriented teaching, teaching is by research, and knowledge is the outcome. Research-oriented teaching has three modes: problem-solving; independent research; and projects - the algorithm course is mainly problem-solving. The specific process in the algorithm course is: teachers create problem situations → students ask questions → gather scientific facts → explore problem-solving approaches → draw conclusions → apply new knowledge.

For example, when explaining backtracking, the double chromosphere (a Chinese lottery) can be taken as an example. Students are told to apply rules of the double chromosphere and use computers to output all combinations. Teachers guide students toward identifying all possible solutions. Then, teachers introduce the idea of backtracking, which compresses the search space. Students, then, design algorithms and implement them and compare running times. Finally, students reach the conclusion that this method can be applied to solving the eight-queens chess problem or the colouring problem.

Reforming the Assessment Methods and Criteria

In traditional teaching, students learn from teachers and textbooks. But in this mode, students are less creative, do not have to identify problems and ask questions, and there is a lack of practical problem-solving exercises. This results from the assessment in the form of a closed book examination, which needs reform. The reforms of the assessment of algorithm design and analysis are as follows:

- Assessment reform: process evaluation and results assessment are equally important. Process evaluation includes classroom performance, homework and experiments. The results assessment includes mid-term and final examinations.
- Assessment criteria reform: in paper examinations, place more emphasis on innovative ideas in solving problems.

REFORM THE TEACHING OF THE ALGORITHM DESIGN AND ANALYSIS COURSE

Teaching involves the tools, media and devices used to pass information. Teaching uses various means and technologies to convey information: language, books, printed materials, electronic audio-visual equipment and multimedia network technology [4].

On-line Educational Platform to Supplement Classroom Teaching

With the rapid development of computer and multimedia technology, a new model of on-line education based on constructivist learning theories as a complement to traditional teaching models based on behavioural and cognitive learning theories has attracted more attention. The algorithms course has been set up as an on-line course using the open-source platform, Claroline. This involves course profiles, lesson plans, lecture notes, course links, learning paths, jobs, study and discussion and collaboration resources.

Blackboard and Multimedia

The blackboard and multimedia courseware in teaching have their own advantages and are complementary. The blackboard is flexible and allow teachers to improvise, but multimedia is able to provide a richer audio-visual environment, so that the teaching content can be more intuitive. In the algorithms course, blackboard-based teaching is used for most of the content supplemented by multimedia courseware for some abstract problems, e.g. the shortest path graph search, graph traversal and various sorting processes. Multimedia presentations facilitate students' understanding of complex algorithms.

CONCLUSION

The focus on teaching algorithm design and analysis should be on the basic theory and methods, so that students master the algorithms, while at the same time improving their ability to analyse and solve problems. But, there is also a need to strengthen students' programming ability by implementing algorithms. The algorithms course involves a wide field of knowledge that is rapidly changing. This makes a high demand on teachers to keep up with the latest developments and to continue to improve the teaching of algorithms.

ACKNOWLEDGEMENTS

This research is supported by the Eleventh Five-Year Plan issues of Hubei Province (2010B115); the National Science Foundation of China (No.61370092, No.61370223); the National Science Foundation of Hubei (No. 2013CFC005); and the Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410).

REFERENCES

1. Hu, F. and Wang, G., Teaching mode exploration on algorithm analysis and design. *Contemporary Educational Theory and Practice*, **12**, 72-74 (2011).
2. Yu, X., Cui, G. and Zou, H., *Basis of Computer Algorithm*. Wuhan: Huazhong Science and Technology University Press (2006).
3. Cao, J., Rational thinking on research-based teaching in university. *Modern University Educ.*, **6**, 111-112 (2002).
4. Wang, X. and Huang, X., Problems and improvement measures of algorithm design and analysis in undergraduate teaching. *Higher Educ. Research*, 29, **2**, 43-46 (2012).