

Research and Practice on Algorithm Analysis and Design Course Teaching Based on Online Judge

Duanyang Zhao, Qingxiang Xu
 Zhijiang College of Zhejiang University of Technology
 Hangzhou, China
 {sunny, xqx, }@zjc.zjut.edu.cn

Abstract—This paper describes the algorithms analysis and design course that uses ACM-ICPC programming competition problems as the basis for the lectures, homework and exams. Programming contest has certain timeliness, openness and objectivity of evaluation. By taking part in programming competitions students can improve solving problems and programming skills. The course of algorithm analysis and design presents the solution strategies of a number of specific issues (such as the sorting problems, search problems, path problems, combinatorial problems, etc.) so that students can master the algorithm design strategies and analysis methods. We prepare many problems for each issue in the online judgment system (OJS). By using OJS, we carry out all experiments and examinations in the specified times. After solving some problem, students submit the code to OJS through browser. We can output the student codes to further analysis in the background management system. Online judge system provides an open, independent teaching platform for students to learn the course knowledge. With OJS students can submit their code at any time and any where.

Keywords- *algorithm analysis and design course; open teaching platform; online judge system (OJS); programming competition*

I. INTRODUCTION

Currently, educational tendencies are centered on the student's point of view rather than the instructor's. The new methods should stimulate students' interest and offer appealing material, fair assessment and relevant feedback [1].

As computer and Internet technology developed, computer aided education and network education have caused a profound revolution in the mode and the notion of education. Computer technology reduces the teachers' workloads and provides a more extensive form of education methods. More and more students can receive higher quality education through Internet [2, 3].

Online Judge System (OJS) is the core of collegiate programming competition. The greatest feature of OJS is that it can automatically test source codes online. Now the system supports three programming languages: C, C++ and Pascal. Students can write their programs to solve a problem in the OJS in any of the languages and then submit source codes to the system. The submitted source code is saved in a file, and a corresponding compiler is selected to compile the source code into an executable program. The executable program will be executed several times according to a number of input/output data pairs pre-prepared for the problem. At each run of the program, one input data is fed to the program and

the program's output are compared with the corresponding pre-prepared output data. If the two outputs is the same, and the time limit and memory limit are not broken, then the program is proved to be accepted for the input/output pair. When the program is proved to be accepted for all input/output pairs, then the program is accepted by the system. If any error happens during the process, the program will be rejected as "Wrong Answer", "Time Limit Exceed", "Memory Limit Exceed", "Runtime Error" and "Compile Error" correspondingly [4].

The paper introduces an innovative experience with the course of "Algorithm analysis and design" with a web-based automatic OJS. The goal of the study is to analyze the effect of this novel teaching methodology on the promotion of student programming skills. Our results show the viability of the proposal and a high capacity to generate motivation and enthusiasm among students.

II. CHARACTERISTIC OF THE ALGORITHM ANALYSIS AND DESIGN COURSE

The students major in computer science and technology specialty, who want to be programmers with strong programming capabilities, have to master the knowledge of high level programming language, data structures, algorithm design strategies and design patterns, software architecture and development method. Algorithm analysis and design course is the core curriculum of programming, mainly introduces the common strategies and complexity analysis of the algorithms. The students learn from the course how to improve their ability for analyzing and solving problems and how to program for large scale software systems [5].

The course is mind and hand, comprehensive, systematic and widespread. Its contents include some common strategies in algorithm design, such as the analysis of algorithm efficiency, divide and conquer method, greedy method, dynamic programming, branch and bound method, backtracking method, approximation and probabilistic algorithms, and some programming methods, such as sorting, searching, graph theory, geometry, composition and numerical calculation problems. The course emphasizes the ideas and the design of algorithms, while we have to pay more attention to application of the algorithms, to teach students how to use computers to solve problems creatively and to foster student ability to independently analyze and solve problems.

In many universities, the teaching method of the course has been still the traditional method, i.e. teacher-centered teaching method. The teachers usually explain to students the

classical algorithms based on classical data structures and mathematical models [6]. Exercises of the course are in the verification algorithms, and neglect the efficiency of the algorithms, the size of test data and the actual scenarios. Learning of students is mainly focus on understanding and memory algorithms and can not be flexible use of algorithms. In the practice session, the students are seriously used to copy the problem codes so that it is difficult to achieve the training effect and to effectively improve student practical ability.

III. PEDAGOGICAL PRINCIPLES IN THE COURSE

In order to reform traditional teaching patterns, we have to adopt a new learning paradigm. The four pedagogical principles underpinning this novel paradigm, following some recommendations by other authors [7], are:

1) *Motivation*. With suitable dropout rates, increasing the motivation of students is clearly essential. By motivation we mean the desire to learn new concepts and methods, and to put them into practice.

2) *Active learning*. The students have to be involved in, and aware of, their own learning process. An active methodology, where students are not mere spectators, is necessary to achieve real and long-lasting learning.

3) *Autonomous work*. We believe that the best way to learn computer programming is by programming. A simple memorization of concepts is nearly useless. The students have to reflect on algorithmic problems by themselves and autonomously create their programs.

4) *Feedback of the learning process*, from the point of view of students. The method should provide students with a continuous evaluation on how well they are doing.

All of these principles require more participatory activities, both in class and out of class. It is in these situations where e-Learning tools can produce a great benefit, not by substituting the teachers, but by helping them to control and evaluate the activities.

IV. THE METHODOLOGY BASED ON OJS

By upholding the pedagogical principles proposed above, we have designed four different ways of using the OJS in the course [7]:

1) *Independent problems*. In the OJS, more than three hundred problems are proposed to the students. There are four volumes. Each volume has one hundred problems except last one. The example of some problems is shown in Table I. The problems are independent of each other, and with different levels of difficulty. The students are expected to select and solve some of them, not necessarily all. Problems can be grouped by category, in such a way that each category illustrates a programming technique discussed in class.

2) *Dependent problems*. This case is preferable when the objective of an activity is to develop a longer and more complex programming project. The project is divided into smaller and consecutive sub problems; each of them is

described as a problem in the judge. The students have to solve all the problems in the given order. In the itinerary, a number of programming techniques can be illustrated. This kind of activities can normally take several months.

TABLE I. THE EXAMPLE OF SOME PROBLEMS

ID	Title	Ratio(AC/submit)	Difficulty
.....			
1339	Arctic Network	56%(5/9)	44%
1340	Addition Chains	25%(1/4)	75%
1341	Binomial Showdown	20%(4/20)	80%
1342	Tree Recovery	100%(3/3)	0%
1343	Do the Untwist	44%(68/155)	56%
1344	Parencodings	58%(69/118)	42%
1345	Number Game	54%(54/100)	46%
1346	The Worm Turns	73%(52/71)	27%
1347	Undercut	60%(50/84)	40%
1348	Currency Exchange	59%(53/90)	41%
1349	Round and Round We Go	61%(54/88)	40%
.....			

3) *Contest style*. Contrary to the other cases, where activities are carried out online, here the presence of the students is required. A set of at most 4-9 problems is given to the participants. They have to try to solve as many problems as possible, and as fast as they can. The contest can take between 2 and 4 hours.

4) *Designing problems*. This is the most creative type of activities. The students have to create a problem with the format of the judge: problem description, source code to solve it, input cases, and expected outputs. The ability of the students to produce original and relevant problems is evaluated.

Each kind of activity is specially focused on a particular cognitive level. For example, the first type is centered on application abilities; the second type tries to improve synthesis skills; the third type aims at developing knowledge and evaluation; and the last type focuses on comprehension and analysis [8, 9].

V. THE FRAMEWORK OF THE OJS

The function of OJS is very useful to algorithm analysis and design course. The students can submit their source codes at any time and get the result quickly. The teacher workloads in checking the source codes can be lighten significantly and they have more time to communicate with their students and answer their questions. The OJS as a smart tool is a good helper in the course.

The OJS is divided into foreground web pages and background management system. With the background management platform, teaches can manage the website contents by the browser.

A. Foreground User Interface

The framework of foreground user interface is shown in Fig. 1, including seven columns.

1) *Problems and Submit Codes*. Students can browse all problems volume by volume, or search some problem by key words in titles. If a student selects a problem and prepare the relevant code, he/she can submit the code by browser.

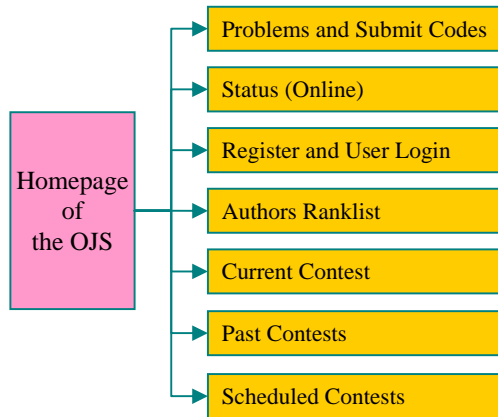


Figure 1. The framework of foreground user interface

2) *Status (Online)*. Once student submitted the code, he/she can view the status that code run. There are six status:

- Accepted

Student program must read data from standard input according to the specification of the contest question. OJS will test it with background secret input. If its output matches the judge output the program will be judged correct.

- Wrong Answer (WA)

If its output does not match what the judges expect, it will get wrong answer notification. Generally, wrong answer occurs because students either have misunderstood the problem, missed a trick in the question, unchecked the extreme conditions or are simply not experienced enough to solve the problem. Problems often contain tricks that are missed by not reading the problem statement very carefully.

- Presentation Error (PE)

This error occurs when student program produces correct output for the judge secret data but does not produce it in the correct format.

- Runtime error (RE)

This error indicates that student program performs an illegal operation when run on judge input. Some illegal operations include invalid memory references such as accessing outside an array boundary. There are also a number of common mathematical errors such as divide by zero error, overflow or domain error.

- Time Limit Exceeded (TLE)

In a contest, the judge has a specified time limit for every problem. When student program does not terminate in that specified time limit student get this error. It is possible that students are using an inefficient algorithm, e.g., trying to find the factorial of a large number recursively, or perhaps there has been a bug in student program producing an infinite loop. Another common error is for student program to use inappropriate input format.

- Maximum Limit Exceeded (MLE)

The maximum memory allowed on the OJS is 32767MB, which includes memory for global variables, the heap, and the stack. If student program tried to use more memory than the judge default settings this error occurs.

3) *Register and User Login*. If a student submit the code for specified problem, he/she has to registered and logged.

4) *Authors Ranklist*. The numbers of solved problems of all users sort list, as descending order by the numbers, as shown in Table II. It encourages student enthusiasm each other for more problem solved.

TABLE II. AUTHORS RANKLIST IN THE OJS

No.	User ID	Solved Problems	Submit numbers	Ratio (AC/submit)
1	bearfly	239	1012	23%
2	fish1210	221	264	83%
3	d518	202	536	37%
4	sunny	196	795	24%
5	wswyb001	194	225	86%
6	yes	177	499	35%
7	yeyoulong	167	266	62%
			

5) *Current Contest*. The contest is running now, and its webpage has four menus: Problems, Standing, Status and Statistics.

- Problems

An example of the problems list is showed in Table III.

TABLE III. THE PROBLEMS IN A CONTEST

Problem Id	Title
1344 Problem A	Do the Untwist
1345 Problem B	Parencodings
1346 Problem C	Number Game
1347 Problem D	The Worm Turns
1348 Problem E	Undercut
1349 Problem F	Currency Exchange
1350 Problem G	Round and Round We Go

- Standing

The numbers of solved problems of all users in the contest sort list, as descending order by the numbers. An example of standing is showed in Table IV.

TABLE IV. THE STANDING IN A CONTEST

Rank	Nick Name	Accepts	Penalty	A	B	G
1	d522	7	2:15:23	0:06:12	0:12:35		0:21:21
2	d303	7	3:00:14	0:07:12	0:12:32		0:29:01
3	d621	7	4:04:55	0:09:41	0:15:52		0:46:40
4	d624	6	7:24:16	0:46:17	0:41:54		
5	d626	6	8:24:16	0:53:18	0:55:07		
6	d222	6	9:17:55		0:59:05		1:16:28
7	d502	5	3:48:43	0:16:00	0:24:25		0:59:21
						

- Statistics

For each problem, we make a statistic for the numbers of accepted (AC), presentation error (PE), wrong answer (WA), time limit exceed (TLE), and others, as shown in Table V.

TABLE V. THE STATISTICS IN A CONTEST

	AC	PE	WA	TLE	CE	Total	G++	GCC
A	67		37			35	154	143	11
B	68	7	24			12	117	107	10
C	53	12	23	2		7	99	90	9
D	51	2	12	2		3	70	68	2
E	49	7	19			7	83	82	1
F	52		18			15	89	87	2
G	49	6	13			6	81	70	11
Total	389	34	146	4		85	693	647	46

6) *Past and Scheduled Contests.* The contests have past or are scheduled for the future. For scheduled contests, students can not view in details.

B. Background Management

Teachers can manage the OJS with background management system, which framework is shown in Fig. 2, including four menus.

1) *Add a Problem.* When adding a problem, teacher has to input problem title, time limit, memory limit, problem description, input and output specifications, sample input and output data, hint.

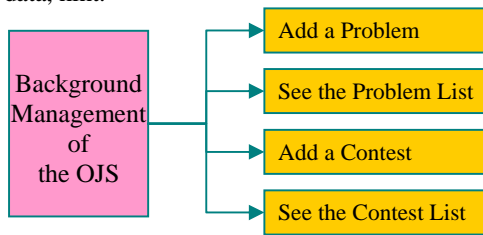


Figure 2. The framework of background management system

2) *See the Problem List.* All problems list. For each problem, teacher can edit or delete it.

3) *Add a Contest.* When adding a contest, teacher has to input the contest title, description, start time and end time, then selects problems for the contest.

4) *See the Contest List.* The scheduled contests list, from which teacher can view in detail, and add, delete or update some problems.

VI. EVALUATION OF THE OJS

A histogram of the problem numbers solved by each student is shown in Fig. 3 [7].

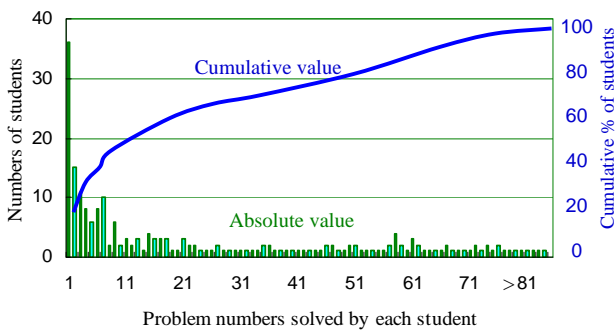


Figure 3. Problem numbers by each student, cumulative values

The results obtained after the application of the judge-based methodology are very promising. “Authors Ranklist” plays a fundamental role in motivating students to solve more problems, faster and efficiently. They want to figure up in the ranking. Since submissions are public, students do not have to feel frustrated when they receive a “wrong answer”. They can see that their classmates are going through the same troubles, and that it is a part of learning. When students solve the proposed problems, they play a leading role in their own learning.

VII. CONCLUSIONS

The innovative experience in algorithm analysis and design course that we have described is excellent. With the OJS, the proposed strategy has achieved its primary goal of improving student programming skills. As a consequence, the pass rate increased. On average, the final grades of the students are higher with the new methodology; however, the high failure rate remains a challenge to be faced in the future.

This approach improves self-assessment skills and encourages students to work independently. Students are more involved in their learning, contributing to a strong foundation for the student’s life-long learning. We believe that the proposed methodology can be easily applied to other courses, particularly in those which include programming assignments, such as C/C++ programming, Data structure.

Two major aspects remain to be improved in the future: the feedback provided by the OJS, and plagiarism detection. We are working on extensions of the OJS to provide detailed feedback in case of “Wrong Answer” and “Runtime Error”, and design and develop the plagiarism detection system.

ACKNOWLEDGMENT

The work has been supported by the university elaborate course of *algorithm analysis and design* of Zhejiang University of Technology, Zhejiang province, China (2010).

REFERENCES

- [1] Zhao Duanyang, Xu Qingxiang, Feng, Zhiling. “The research and practice of individualistic education for students major in computer science in independent colleges,” Proc. 2nd international workshop on education technology and computer science, ETCS 2010, IEEE Press, Mar. 2010, v1, pp. 789-792.
- [2] Duanyang Zhao, Qingxiang Xu, Wuheng Zuo. “Research and practice in the curriculum system for majors in software engineering,” Proc. 2nd international workshop on education technology and computer science, ETCS 2010, v1, pp. 417-420.
- [3] Duanyang Zhao, Qingxiang Xu, Wuheng Zuo. “The research and practice of computer teaching at independent colleges for students at different levels,” Proc. 2009 WASE international conference on information engineering, ICIE 2009, v1, pp. 20-23.
- [4] Yingwei Luo, Xiaolin Wang, Zhengyi Zhang. “Programming grid: a computer-aided education system for programming courses based on online judge,” Proc. the 1st acm summit on computing education in china on first acm summit on computing education, SCE '08.
- [5] Chunming Yang, Niannian Chen. “The teaching exploration and practice of algorithm analysis and design base on programming contest,” in Chinese. Computer education. Tsinghua University. Beijing, 2009, (20), pp. 146-149.
- [6] Qun Dai. “Teaching research on the degree course of algorithm analysis and design course,” in Chinese. Computer education. Tsinghua University. Beijing, 2009(18), pp. 70-72.
- [7] Ginés García-Mateos, José Luis Fernández-Alemán. “A course on algorithms and data structures using on-line judging,” Proc. the 14th annual ACM SIGCSE conference on innovation and technology in computer science education, ITiCSE '09.
- [8] John Paxton. “Programming competition problems as a basis for an algorithms and data structures course,” Journal of computing sciences in colleges. Bozeman, 2007, (23)2.
- [9] John Paxton, Brendan Mumey. “Teaching advanced problem solving: implications for the CS curriculum,” Proc. of the fourteenth annual consortium on Small Colleges Southeastern conference, CCSC '00.