Education Theory: Teaching and Learning (ISSN Online: 2771-9030)
June 2022 Vol.1, No.3, pp.108-112
DOI: https://doi.org/10.55571/ettl.2022.06021

4D Publishing Group

Online Judge Systems and Case Teaching Used in Programming Courses

Yong-hui Wu
Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China

Email:

yhwu@fudan.edu.cn

Abstract. The key to programming courses, including programming languages, data structure, algorithm design and analysis, and so on, is to polish students' programming skills solving problems. The current programming courses in universities are mainly classroom-teaching models, and students' skills solving problems by programming can hardly be polished. Polishing students' programming skills solving problems is implemented by teaching material construction and curriculum construction. In teaching material construction, the book series "Collegiate Programming Contests and Education" focus on polishing students' programming skills solving problems in a systematic way, based on programming contests' problems. Experiments, consisting of programming contest problems and their analysis and solutions, are basic units for the book series. The curriculum construction's guiding ideology is programming is a technology. In curriculum construction, first, the case teaching is widely used based on experiments. Students are put into a case of a problem description, apply knowledge that they have learned, think how to solve the problem, and propose the algorithm. And after students design the algorithm solving the problem, they need program and debug to pass all test cases within the time and memory limit. Second, informatization technologies, such as online judge systems, and virtual online contests based on online judge systems, are integrated into courses. Online judge systems are online systems to test programs whether is correct or not in programming practice. These systems can compile and execute programs, and test programs with the input for test data. Online judge systems are the platform on which students polish their programming skills. It has been proved by practice that these works can polish students' programming skills solving problems efficiently.

Keywords. Programming Course, Programming Skill, Curriculum Construction, Case Teaching, Online Judge System, Virtual Programming Contest

1. Introduction

In the current information society, programming technology becomes a technology that whole people should master. However, programming courses are mainly classroom-teaching models, and students' skills solving problems by programming can hardly be polished.

Education Theory: Teaching and Learning (ISSN Online: 2771-9030)
June 2022 Vol.1, No.3, pp.108-112
DOI: https://doi.org/10.55571/ettl.2022.06021

Programming contests are contests solving problems by programming. There are a lot of programming contest problems which can be gotten from Internet. And these programming contest problems can be used in education, to polish students' programming skills solving problems greatly.

When we evaluate a person's professional ability, we need evaluate the person's two aspects: First, his knowledge system; that is, the knowledge that he can use to solve practical problems; and second, his mode of thinking; that is, the strategies that he will take when he solve non-standardized problems. The programming knowledge system can be summarized as a famous formula: "Algorithms + Data Structures = Programs". It is also the core of the knowledge system for computer science. Programming strategies solving problems are strategies for data modeling and algorithm design. If a problem is solved by some advanced data structures and optimized algorithms, programming strategies should be taken.

In teaching material construction, the book series "Collegiate Programming Contests and Education" based on programming contest problems from all over the world has been compiled and published in simplified and traditional Chinese and English: the former by respective publishers of mainland China and Taiwan, and the latter, by CRC Press [1-11]. The book series can be used not only in the systematic programming contest training for contestants, but also in programming courses and experiments in universities, to polish students' programming skills better. Four books constitute the book series: Data Structure Practice, Algorithm Design Practice, Programming Strategies Solving Problems, and Preliminary Programming Practice. The first two books are used to systematically polish students' programming skills solving problems by data structure and algorithm. The formula "Algorithms + Data Structures = Programs" summarizes the core of the knowledge system for computer science. The third book "Programming Strategies Solving Problems" focuses on solving problems by advanced data structures and optimized algorithms. And the fourth book "Preliminary Programming Practice" is for programming beginners.

The paper introduces the curriculum construction based on the book series, including case teaching and online judge systems [12-16].

2. Case Teaching: The Teaching Model

2.1. Experiments: Based on Programming Contest Problems

Experiments, based on programming contest problems, are basic units for the book series.

In an experiment, first, the knowledge point is introduced; second, related programming contest problems and their analysis and solutions with detailed annotations are showed.

In a programming contest problem, there are 3 parts: the problem description, specifications for input and output, and a sample input and output.

An experiment for Huffman Tree is as follow.

In a binary tree there are n leaves with weights, where the weight of the k-th leaf is w_k, and the length of the path from the root to the kth leaf is p_k. Then w_k*p_k is the length of weighted path for the k-th leaf, $1 \le k \le n$. The binary tree whose the sum of lengths of weighted paths is minimal is called a Huffman tree. That is, if $L=\sum_{k=1}^{n} w_k p_k$ is minimal, the binary tree is a Huffman tree.

Given n nodes whose weights are $w_1, w_2, \ldots, w_n$ respectively, the process constructing a Huffman tree is as follows.

First, n nodes constitute a set of n binary trees $F=\{T_1, T_2, \ldots\ldots, T_n\}$, where T_i has only one node whose weight is w_i, $1 \le i \le n$.

while (F is not a tree)
{ Replace the rooted trees T and T' of least weights from F with a tree having a new root that has T as its left subtree and T' as its right subtree;
The weight of the new tree = the weight of T + the weight of T';
}

A Huffman tree is a complete binary tree. In a Huffman tree, if there are n leaves, there are 2n-1 nodes. Constructing a Huffman tree is a greedy method, each time two trees with least weights are

Education Theory: Teaching and Learning (ISSN Online: 2771-9030)
June 2022 Vol.1, No.3, pp.108-112
DOI: https://doi.org/10.55571/ettl.2022.06021

selected. Therefore, a min heap is used to store roots of trees when a Huffman tree is constructed.

【Fence Repair】

Farmer John wants to repair a small length of the fence around the pasture. He measures the fence and finds that he needs N ($1 \le N \le 20,000$) planks of wood, each having some integer length L_i ($1 \le L_i \le 50,000$) units. He then purchases a single long board just long enough to saw into the N planks (i.e., whose length is the sum of the lengths L_i). FJ is ignoring the "kerf", the extra length lost to sawdust when a sawcut is made; you should ignore it, too.

FJ sadly realizes that he doesn't own a saw with which to cut the wood, so he mosies over to Farmer Don's Farm with this long board and politely asks if he may borrow a saw.

Farmer Don, a closet capitalist, doesn't lend FJ a saw but instead offers to charge Farmer John for each of the N-1 cuts in the plank. The charge to cut a piece of wood is exactly equal to its length. Cutting a plank of length 21 costs 21 cents.

Farmer Don then lets Farmer John decide the order and locations to cut the plank. Help Farmer John determine the minimum amount of money he can spend to create the N planks. FJ knows that he can cut the board in various different orders which will result in different charges since the resulting intermediate planks are of different lengths.

Input

Line 1: One integer N, the number of planks;

Lines 2..N+1: Each line contains a single integer describing the length of a needed plank.

Output

Line 1: One integer: the minimum amount of money he must spend to make N-1 cuts.

Sample Input	Sample Output
3 8 5 8	34

Hint

He wants to cut a board of length 21 into pieces of lengths 8, 5, and 8.

The original board measures 8+5+8=21. The first cut will cost 21, and should be used to cut the board into pieces measuring 13 and 8. The second cut will cost 13, and should be used to cut the 13 into 8 and 5. This would cost 21+13=34. If the 21 was cut into 16 and 5 instead, the second cut would cost 16 for a total of 37 (which is more than 34).

Source: USACO 2006 November Gold

IDs for Online Judge: POJ 3253

Analysis

Because each cut produces two planks of wood, the process cutting planks can be represented as a binary tree. The initial single long board is as the root, and the length of the board is its weight; N planks are as N leaves, where the weight of the i-th leaf is the length of the i-th plank w_i, and the length from the root to leaf p_i is the number of cuts producing the i-th plank. Based on the problem description, the cost for cutting the i-th plank is p_i*w_i, $1 \le i \le n$. Obviously the total cost is $\sum_{k=1}^{n} w_k p_k$. Therefore calculating the minimal charge to cut a piece of wood is to calculate a Huffman tree. The process is as follows.

A min heap is constructed based on lengths of N planks. Each time roots of the heap are deleted twice. The two deleted roots' weights are a and b respectively. A new node whose weight is (a+b) is inserted into the min heap. And the cost ans increases (a+b). Repeat the process until there is only one node in the min heap. At that time ans is the minimal cost.

Program (Omitted)

2.2. Case Teaching based on Experiments

Experiments are based on programming contest problems. Therefore the teaching model for courses is case teaching. The process is as follow.

First, the knowledge point for an experiment is introduced. Then, students read the related programming contest problem and are put themselves into the situation of the problem description, apply the knowledge point that they have just learned, and design the algorithm solving the problem. Finally, students program and debug to pass all test cases within the time and memory limit.
Such a case teaching combines practice with thinking, stimulates students' desire for knowledge, and deepens their understanding knowledge. Therefore such a process promotes teaching innovation and course construction based on programming contest problems.

3. Informatization Technologies

3.1. Online Judge Systems

Online judge systems are used to test programs whether is correct or not. In programming practice, students submit their programs to systems through Internet. After programs are received, online judge systems compile and execute programs, and test programs with the input for test data. Programs read input from standard input and write output to standard output. Programs are run with restrictions, such as time limit, memory limit, and so on. The output of programs are compared with the output for test data. Then systems return the result.

Online judge systems are platforms on which students polish their programming skills. Using online judge systems is the foundational informatization technologies for polishing students' programming skills.

3.2. Virtual Programming Contest

The curriculum construction's guiding ideology is programming is a technology. Therefore, for polishing students' programming skills, first, students must practice, practice, and practice; that is, students are required to solve programming contest problems; and the more, the better. The genuine knowledge comes from practices, and practices polish students' programming skills solving problems. Second, students should practice in a systematic way; that is, students are required to solve programming contest problems not only based on the syllabus, but also with the help of problems' analyses, test data, and solutions with detailed annotations. It enables students to construct their own knowledge systems solving problems by programming step by step. Third, students must practice under pressure. Homework and the examinations are set as virtual online programming contests. Students solve problems under pressure. Virtual online programming contests make students to polish programming skills efficiently.

There are many online judge systems. Virtual programming contests combine problems from different online judge systems together. Ranks for virtual programming contests are used to evaluate students' results for their programming skills. Teachers can manage students' homework and examinations through virtual programming contests.

4. Conclusion

Online judge systems and case teaching used in programming courses are introduce. It has been proved by practice that the works can polish students' programming skills solving problems efficiently. Students not only get better achievements in programming contests, but also show better programming and thinking ability. In the future, online programming courses cross region will be developed and popularized.

References

[1] Yonghui Wu, Jiande Wang. Algorithm Design Practice : for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version). Beijing: China Machine Press. 2020.
[2] Yonghui Wu, Jiande Wang. Algorithm Design Practice: for Collegiate Programming Contest and Education (English Version). Orlando: CRC Press. 2018.
[3] Yonghui Wu, Jiande Wang. Data Structure Practice: for Collegiate Programming Contest and Education (English Version). Orlando: CRC Press. 2016.

[4] Yonghui Wu, Jiande Wang. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Traditional Chinese Version). Taipei: GOTOP INFORMATION INC. 2017.
[5] Yonghui Wu, Jiande Wang. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version). Beijing: China Machine Press. 2016.
[6] Yonghui Wu, Jiande Wang. Programming Strategies Solving Problems: for Collegiate Programming Contest and Education (Simplified Chinese Version). Beijing: China Machine Press. 2015.
[7] Yonghui Wu, Jiande Wang. Programming Strategies Solving Problems: for Collegiate Programming Contest and Education (Traditional Chinese Version). Taipei: GOTOP INFORMATION INC. 2015.
[8] Yonghui Wu, Jiande Wang. Algorithm Design Experiment: for Collegiate Programming Contest and Education (Simplified Chinese Version). Beijing: China Machine Press. 2013.
[9] Yonghui Wu, Jiande Wang. Solutions And Analyses To ACM-ICPC World Finals(2004-2011) (Simplified Chinese Version). Beijing: China Machine Press. 2012.
[10] Yonghui Wu, Jiande Wang. Data Structure Experiment : for Collegiate Programming Contest and Education (Simplified Chinese Version). Beijing: China Machine Press. 2012.
[11] Yonghui Wu, Jiande Wang. Data Structure Experiment : for Collegiate Programming Contest and Education (Traditional Chinese Version). Taipeu: GOTOP INFORMATION INC. 2012.
[12] Yonghui Wu. Cooperating Programming Contest Training with Education. Competitive Learning Institute Symposium (CLIS) 2019. https://ciiwiki.ecs.baylor.edu/index.php/ICPC_CLIS_2019,_Porto,_Portugal, Porto, Portugal.
[13] Yonghui Wu. The Book Series “Collegiate Programming Contests and Education”. Competitive Learning Institute Symposium (CLIS), 2018, https://ciiwiki.ecs.baylor.edu/index.php/ICPC_CLIS_2018,_Beijing,_China, Beijing, China.
[14] Yonghui Wu, Jingshan Yu, Xuefeng Jiang, Sheng-Lung Peng. Programming Training League: A System Organizing Programming Training Cross Region. Competitive Learning Institute Symposium (CLIS), 2018, https://ciiwiki.ecs.baylor.edu/index.php/ICPC_CLIS_2018,_Beijing,_China, Beijing, China
[15] Yonghui Wu. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version). Fudan Mooc, https://mooc1-1.chaoxing.com/course/215508752.html, 2019, Shanghai, China.
[16] Yonghui Wu. The Implementation for Polishing Students’ Programming Skills Solving Problems. Proceedings of the 2021 International Conference on Diversified Education and Social Development. Beijing: Atlantis Press. 2021:92-97.