Advances in Social Science, Education and Humanities Research, volume 569
Proceedings of the 2021 International Conference on Diversified Education and Social Development (DESD 2021)

The Implementation for Polishing Students' Programming Skills Solving Problems

Yonghui Wu[*]

Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University
[*]Corresponding author. Email: yhwu@fudan.edu.cn

ABSTRACT
The key to programming education is to polish students' programming skills solving problems. The current programming courses in universities are mainly classroom-teaching models, and students' skill solving problems by programming can hardly be polished. Polishing students' programming skills solving problems is implemented by teaching material construction, curriculum construction and programming training system cross region. 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. In curriculum construction, the case teaching is widely used based on experiments; and informatization technologies, such as online judge systems, and virtual online contests, are integrated into courses. A programming training system cross region, ICPC Asia Training League and ICPC International Joint Labs for Programming Technologies, is set up. It has been proved by practice that these works can polish students' programming skills solving problems efficiently.

Keywords: *International Collegiate Programming Contest (ICPC), Programming technology, Programming strategy, Case teaching, Informatization technology, Online judge system.*

1. INTRODUCTION

As informatization progresses in society, programming technology plays more and more important role. For the information society, programming technology is the implementation technology. Some new information technologies, such as Big Data, AI, block chain, and so on, are implemented by programming. Programming technology is becoming the technology that whole people should master.

The current programming courses in universities, programming languages, data structure, algorithm analysis and design, and so on, are mainly classroom-teaching models, and focus on teaching contents. Therefore students' skill solving problems by programming can hardly be polished when they learn these courses.

Programming contests are contests solving problems by programming. ACM International Collegiate Programming Contest (ACM-ICPC), founded in 1970s, is the oldest, largest, and most prestigious programming contest in the world. Alongside, some other international programming contests, such as Google Code Jam, TopCoder Open Algorithm, Facebook Hacker Cup, Internet Problem Solving Contest (IPSC), and so on, are also held every year.

A large number of programming contest problems have been accumulated these years. And these problems can be used not only to train programming contestants. If these problems are used in education, students' programming skills solving problems can be improved greatly.

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][2][3][4][5][6][7][8][9][10][11]. It is the first book series in the world that 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 skill better.

The paper introduces the book series and the curriculum construction and the programming training system cross region based on the book series.

2. THE TEACHING MATERIAL CONSTRUCTION: THE BOOK SERIES "COLLEGIATE PROGRAMMING CONTESTS AND EDUCATION"

2.1 The Knowledge Structure for the Book Series

If we intend to 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.

Now a large number of programming contests' problems from all over the world can be gotten from Internet. In programming contest training, programming contestants can analyze and solve these problems to polish their programming skills. And if these problems are used in education, students' programming skills solving problems can be improved greatly.

The book series "Collegiate Programming Contests and Education" is based on above opinions. Four books constitute the book series: Data Structure Practice, Algorithm Design Practice, Programming Strategies Solving Problems, and Preliminary Programming Practice.

The programming knowledge system can be summarized as the famous formula: "Algorithms + Data Structures = Programs". Therefore, the first two books are used to systematically polish students' programming skills solving problems by data structure and algorithm. The outlines for the first two books are based on outlines of data structure and algorithm respectively. That is, the layout for chapters and sections are based on the knowledge system for data structure and algorithm. Strategies solving problems are strategies for data models and algorithm design. When data models and algorithms for problems are not standard, we need take some strategies to solve these problems. And the third book "Programming Strategies Solving Problems" focuses on solving problems by advanced data structures and optimized algorithms. The fourth book "Preliminary Programming Practice" is for programming beginners.

For the book series, in chapters, several sections are showed as experiments. In these sections, first, the knowledge background for data structure or algorithm is introduced; second, relevant programming contest problems and their analyses and solutions are showed as experiments. Sources and IDs for online judge systems for these problems are also given. Experiments are basic units for the book series.

2.2 The Content of "Data Structure Practice"

The book "Data Structure Practice: for Collegiate Programming Contests and Education" has been published in simplified and traditional Chinese, and English. In its new version, there are 4 parts, 15 chapters, and 306 programming contest problems. It is the first book in the book series.

Part 1 "Fundamental Programming Skills" focuses on experiments and practices for simple computing, simple simulation, recursion and backtracking algorithm, for students just learning programming languages. Part 2 "Experiments for Linear Lists", Part 3 "Experiments for Trees", and Part 4 "Experiments for Graphs" focus on experiments and practices for data structure. Chapters in the book not only covers data structure, but also covers some related advanced data structures in set and graph theory.

In Part 1, first, students begin to learn how to transfer a practical problem into a computing process, implement the computing process by a program, and debug the program to pass all test cases. Then, programming skills for off-line method, precision of real number and dichotomy should also be polished. For recursion, not only experiments for solving problems by recursive algorithms are given. Some data structures, such as trees and binary trees, are in a recursive forms, experiments for solving problems by recursive datum are given, and as the foundation for such recursive datum. For backtracking algorithm, "eight queens" is the foundation problem, and in programming contest problem "The Sultan's Successors", all solutions to "eight queens" should be calculated. And based on the problem, the experiment for the optimization problem solved by backtracking algorithm is showed. Experiments for backtracking algorithm is also the foundation for search technologies.

In Part 2, there are 4 chapters: "Linear Lists Accessed Directly", "Applications of Linear lists for Sequential Access", "Generalized List Using Indexes", and "Sort of Linear Lists". Not only applications for arrays, strings, pointers, stacks, queues and dictionaries are showed, but also experiments for Manacher Algorithm, hash technologies and hash tables are given. And for sorting linear lists, not only using sorting algorithms, but also using sort function in STL are discussed.

In Part 3, there are 3 chapters: "Programming by Tree Structure", "Applications of Binary Trees" and "Applications of Classical Trees". First, tree structures can be used to represent hierarchical structures and union-find sets. Then, applications of binary trees and classical trees are discussed. And experiments for properties of binary trees, traversal of binary trees, binary search trees, binary heaps, Huffman trees, and so on, are showed. And based on it, experiments for quadtrees used to represent two-dimensional spaces, trie trees used for locating specific keys from within a set, Aho-Corasick automaton used for locating elements of a finite set of strings, and some improved BST, such as treap, AVL tree, and splay tree, are showed.

In Part 4, there are 5 chapters "Applications of Graph Traversal", "Algorithms of Spanning Trees", "Algorithms of Best Paths", "Algorithms of Bipartite Graphs and Flow Networks", and "Practice for State Space Search". In chapter "Applications of Graph Traversal". Graphs are used to represent objects and relationships among objects in the real world. First, experiments for BFS and DFS are given, then based on it, experiments for topological sort and connectivity of graphs are showed. BFS and DFS are foundation for many graph algorithms. In "Algorithms of Spanning Trees", not only, experiments for algorithms of minimum spanning trees, Kruskal algorithm and Prim algorithm, are given, but also experiments for maximum spanning trees are given, based on the correctness proof of greedy algorithms. "Algorithms of Best Paths" consists of experiments for Warshall algorithm and Floyd-Warshall algorithm, Dijkstra's algorithm, Bellman-Ford algorithm, and Shortest Path Faster algorithm (SPFA algorithm). In chapter "Algorithms of Bipartite Graphs and Flow Networks", first, theory for Bipartite Graphs and Flow Networks is introduced, then based on the theory, experiments for Hungarian algorithm, Hall's Marriage Theorem, KM algorithm, Flow Networks, and Minimum-Cost Flow Problem, are showed. Finally, search technologies in graphs are discussed. Search technologies are fundamental technologies in computer science and technology. And a search problem can be represented as a state space. In "Practice for State Space Search", first, experiments for constructing a state space tree based on state, successor function, state, and cost are showed. Then, experiments for optimization strategies: branching, memorization, indexing, pruning, bounding, and A* algorithm, are also showed. Finally, experiments for Minimax Algorithm are given.

2.3 The Content of "Algorithm Design Practice"

The book "Algorithm Design Practice: for Collegiate Programming Contests and Education" has been published in simplified Chinese, and English. In its new version, there are 8 chapters, and 314 programming contest problems. The 8 chapters are as follows: "Practice for Ad Hoc Problems", "Practice for Simulation Problems", "Practice for Number Theory", "Practice for Combinatorics", "Practice for Greedy Algorithms", "Practice for Dynamic Programming", "Practice for Advanced Data Structures", and "Practice for Computational Geometry".

Ad hoc means "for the special purpose or end presently under consideration". There are no classical algorithms that can solve these ad hoc problems. Programmers need to design specific algorithms to solve ad hoc problems. There are two strategies to design algorithms for solving ad hoc problems: Mechanism analysis and statistical analysis. There are two strategies to design algorithms for solving ad hoc problems: Mechanism analysis and statistical analysis.

In the real world, there are many problems that we can solve by simulating their processes. Such problems are called simulation problems. For these problems, solution procedures or rules are showed in problem descriptions. Programs must simulate procedures or implement rules based on descriptions. A simulation problem gets harder as solution procedures or rules increases are complicated. It causes the amount of code to grow up and get more illegible.

For Greedy Algorithms and Dynamic Programming, first, experiments for classical problems, such as Knapsack Problem, Task Schedule, Subset Sum, Longest Common Subsequence, Longest Increasing Subsequence, 0-1 Knapsack Problem, and so on, are given. Then, based on it, experiments for Greedy-Choices Based on Sorted Data, Tree-Like Dynamic Programming, Dynamic Programming with State Compression, and so on, are showed.

For mathematic problems, number theory, combinatorics, and computational geometry, experiments are based on mathematic outlines.

2.4 Contents of "Programming Strategies Solving Problems" and "Preliminary Programming Practice"

The third book "Programming Strategies Solving Problems: for Collegiate Programming Contests and Education" has been published in simplified and traditional Chinese. Strategies for data structures consists of "Strategies for Trees", "Strategies for Graphs", and "Construction Strategies for Data Structures". In "Strategies for Trees", experiments for segment trees, spanning trees, dynamic trees, leftist trees, skip lists, and so on, are given. That is, the chapter discusses about solving problems by improved trees. For example, segment trees are used to solve interval problems. "Strategies for Graphs" focuses on solving problems by graph modeling. "Construction Strategies for Data Structures" shows problems are represented and solved by combination of different data structures. Strategies for algorithm includes "Application Strategies for Dichotomy", "Optimization Strategies for Dynamic Programming", "Strategies for Computational Geometry", and "Strategies for Games".

In "Preliminary Programming Practice", there are 6 chapters. The first three chpaters "Input and Output for a program", "Fundamental Programming (I)" and "Fundamental Programming (II)" are experiments for programming languages. The fourth chapter "Preliminary Mathematics Practice" includes experiments for calculus, matrix, and elementary mathematics. Chapter 5 Sorting shows experiments using sorting algorithms, using sort function in STL, and sorting structural array. The last chapter gives experiments for C++ STL.

3. CURRICULUM CONSTRUCTION

3.1 Case Teaching: The Teaching Model

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

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 must program and debug to pass all test cases within the time and memory limit.

The 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. Chapter 5

3.2 Online Judge: The Foundational Informatization Technology

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

3.3. Programming is a Technology: The Curriculum Construction's Guiding Ideology

The curriculum construction's guiding ideology is programming is a technology. Therefore, for polishing students' programming skills, first, students must be required to practice, practice, and practice; that is, students are required to solve programming contest problems; and the more, the better. The genuine knowledge comes from practice, and practice enhances one's ability. Second, students should be arranged to 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; that is, the homework is set as a virtual online programming contest, and students solve problems under pressure. And it enables students to polish programming skills efficiently.

Courses cross region, integrated with case learning and informatization technologies, and based on the book series, run not only in programming training camps, but also as online optional courses cross region[12] [13]. In courses, lectures are given through E-learning system,. In lectures, case teaching is used based on experiments. First, the knowledge background in an experiment is introduced; then students read programming contest problems related to the knowledge background and consider how to solve these problems. Finally, the analysis to these problems are showed. After lectures, virtual online programming contests based on programming contest problems which are related to lectures are set as homework. Virtual online programming contests are based on online judge systems. Students solve problems with the help of test data, solutions and analyses, under pressure.

4. PROGRAMMING TRAINING SYSTEM CROSS REGION

The worldwide scholarly connection for polishing students' programming skills has been set up[14].

The organizational form for the programming training system cross region is based on ICPC Asia Training League and ICPC International Joint Labs for

Programming Technologies. Universities can volunteer to join the league. The league offers online optional courses for programming technologies[15]; organizes programming training camps in vacations to help students train more systematically; organizes practice contests usually; and holds all kinds of official ICPC programming contests, such as university contests, local contests, invitation contests, regional contests, and so on. Now more than 300 universities, from Mainland China, Taiwan, HongKong, and Macau, join the league. And some universities in South Asia and Southeast Asia are observers for the league.

For ICPC Asia Training League, ICPC International Joint Labs for Programming Technologies are as local centers. First, the lab focuses on improving programming courses and experiments in universities, such as programming languages, data structure, algorithm design, and so on, to polish students' programming skills in university education. Second, it is a local programming training center. It holds programming training camps in vacations for specific courses, such as courses for number theory, or game theory. Third, it organizes local examination or programming contests.

Therefore, in the programming training system cross region, ICPC Asia Training League is the foundation, and ICPC International Joint Labs are key points, and promote work in all areas.

5. CONCLUSION

Polishing students' programming skills solving problems is implemented by teaching material construction, curriculum construction and programming training system cross region. It has been proved by practice that the work 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, first, the book series will be improved in practice. Second, the courses cross region based on the book series and informatization technologies will be popularized. Third, the programming training systems cross region, ICPC Asia Training League and ICPC International Joint Labs for Programming Technologies, will be perfected.

REFERENCES

[1] Yonghui Wu, Jiande Wang. Algorithm Design Practice : for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2020.

[2] Yonghui Wu, Jiande Wang. Algorithm Design Practice : for Collegiate Programming Contest and Education (English Version) [M]. Orlando: CRC Press. 2018.

[3] Yonghui Wu, Jiande Wang. Data Structure Practice : for Collegiate Programming Contest and Education (English Version) [M]. Orlando: CRC Press. 2016.

[4] Yonghui Wu, Jiande Wang. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Traditional Chinese Version) [M]. Taipei: GOTOP INFORMATION INC. 2017.

[5] Yonghui Wu, Jiande Wang. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2016.

[6] Yonghui Wu, Jiande Wang. Programming Strategies Solving Problems : for Collegiate Programming Contest and Education (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2015.

[7] Yonghui Wu, Jiande Wang. Programming Strategies Solving Problems : for Collegiate Programming Contest and Education (Traditional Chinese Version) [M]. Taipei: GOTOP INFORMATION INC. 2015.

[8] Yonghui Wu, Jiande Wang. Algorithm Design Experiment : for Collegiate Programming Contest and Education (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2013.

[9] Yonghui Wu, Jiande Wang. Solutions And Analyses To ACM-ICPC World Finals(2004-2011) (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2012.

[10] Yonghui Wu, Jiande Wang. Data Structure Experiment : for Collegiate Programming Contest and Education (Simplified Chinese Version) [M]. Beijing: China Machine Press. 2012.

[11] Yonghui Wu, Jiande Wang. Data Structure Experiment : for Collegiate Programming Contest and Education (Traditional Chinese Version) [M]. Taipeu: GOTOP INFORMATION INC. 2012.

[12] Yonghui Wu. Cooperating Programming Contest Training with Education [EB/OL]. Competitive Learning Institute Symposium (CLIS) 2019. Porto, Portugal.

[13] Yonghui Wu. The Book Series "Collegiate Programming Contests and Education" [EB/OL]. Competitive Learning Institute Symposium (CLIS), 2018, Beijing, China.

[14] Yonghui Wu, Jingshan Yu, Xuefeng Jiang, Sheng-Lung Peng. Programming Training League: A System Organizing Programming Training Cross

Region [EB/OL]. Competitive Learning Institute Symposium (CLIS), 2018, Beijing, China.

[15] Yonghui Wu. Data Structure Practice: for Collegiate Programming Contest and Education (Second Edition) (Simplified Chinese Version) [EB/OL]. JISUANKE(https://www.jisuanke.com/course/1048), 2018, Beijing, China.