Training Plan

Coach - Jiang Shaofeng (姜少峰)

Currently
3rd year PhD supervised by Dr. Hubert Chan
Research interest: theoretical computer science

ICPC
Former ICPC contestant, gold medal in Asia regional contest 2011.

Contact
Homepage: http://i.cs.hku.hk/~sfjiang
Office: CYC LG 101
Email: sfjiang@cs.hku.hk

Our Website

Our website is http://i.cs.hku.hk/~provinci

We post announcements and training materials.

Introduction

What is it?

Not a course. Our interests in it bring us here.
Designed for **new comers**.
Experienced contestants may also benefit.

What do we do?

For me: give you guidance and technical helps.

For you: have passion, be motivated, **work hard**.

Aspects to Be Covered

- Coding skill
- Algorithmic techniques
- Practice contests

Coding Skills

The most basic and important skill.

Strategy:

1. Know a programming language well

- ➢ C++: STL, OOP (C++ Primer)
- ➢ Java: Collections framework, I/O, java.math.*

2. Practice a lot on simulation style questions (e.g. POJ 3699)
3. Implement difficult algorithms and data structures

Material:

1. USACO Training (http://train.usaco.org/usacogate)
2. Understand and implement basic algorithms (e.g. Introduction to Algorithms)
3. Implement algorithms in the algorithmic techniques training.

Algorithmic Techniques

Goal: You know whatever other teams know.

We will cover most algorithms and data structures that may be used in ICPC.

They share something in common!

1. exact and deterministic
2. can be implemented in limited time (e.g. in 300 lines of code)

Fact: they are not cutting edge, and they are not complicated.
Conclusion: They are not too many, and we can cover most of them!

To name a few…

- Dynamic programming. Chain model, tree model, automata model, bit mask.
- Greedy.
- Graph algorithms. Algorithms based on DFS (topological sort, strongly connected components, bi-connected components, etc.), shortest path, network flow, bipartite matching.
- String algorithms. KMP, AC automata, suffix array.

To name a few…

- Data structures. Stack, queue, BST, interval data structures, persistent data structures, data structures for dynamic trees.
- Computational geometry. Basic operations (affine transform, intersection, area, etc.), convex hull related algorithms, sweeping lines.
- Algorithms in number theory. GCD, LCM, primality test, modular equations.
- Counting techniques.
- Advanced divide and conquer.

How?

My side.

- Tell you the topics you need to learn.

Notes: http://i.cs.hku.hk/~provinci/training.html

If we have time, I can talk about each topics

- Leave you practice questions - I will hold lab sessions for it.

Your side.

[Important!] Do the practice, at least what I leave to you.
How much you can learn depends on how well you do the practices.

Practice Contests

- I will arrange practice contests regularly.

More frequently when regional contest is near, and less frequent otherwise.

- You can also find contests held at

❖ Topcoder Arene

❖ Codeforces

❖ Hackerrank

Seeking for help?

- Best way to reach me is email: sfjiang@cs.hku.hk.
- I do appreciate if you can solve the problem yourself.
- If you want to talk to me at my office, drop me an email first.