NOI.PH 2017 Training Week 9

Kevin Charles Atienza

April 2017

Contents

1 Introduction

The in-house training and the selection exam are fast approaching! It would be best to focus the last few days consolidating and applying what you've learned. This week, we will prepare you for the in-house training which will consist mostly of practice/real contests.

I painstakingly compiled 8 four-problem sets, requiring the topics we've covered, for you to practice on this week. I tried to evenly distribute the difficulty and topics, although don't expect it to be *perfect*; some problem sets are probably more data-structure-heavy, others more trick-based, and some sets are probably easier than others.

Although you may solve the problems in any order you want, it is recommended to solve them *per problem set*, and to solve each problem set as a group.

1.1 Scoring

Each problem is worth 1 point, regardless of difficulty. Completing a problem set nets you an additional 1 point. Completing a problem within its target date nets you some bonus.

The score for this week is determined by the number of points over $\max(30, \mathrm{maxscore})$, where $\mathrm{maxscore}$ is the maximum score of any participant, not counting the target bonus.

Please don't waste these prepared problem sets; use them as real practice. Your goal is to maximize your chances to get a medal in the IOI!

1.2 Recommendation

Each set is roughly equivalent to a short contest: 3-5 hours. Although you don't have to follow that strictly, I recommend trying out each set as a single round on its own and timing yourself. (If you do this, please let us know, and also please tell us how you did!) This will give you a rough estimate on how well you could do at the IOI: note that the problems in the IOI will be harder on average. But if you don't meet your set target time, please continue trying to solve them anyway to get points; remember that a full set gives you an extra point.

I *highly* recommend being ahead by 2 or 3 problem sets every day, so if you don't finish a problem set, you'll have time to spare to (maybe) try to get that 3rd or 4th problem (and thus the extra point) by the target date.

Feel free to ask for advice or hints. (Yes, I'll give hints for some problems.)

2 The problem sets

2.1 Problem set HA

Target date: **May 4**

1. Edges in MST: http://codeforces.com/contest/160/problem/D
2. Chef and Prime Divisors: https://www.codechef.com/problems/CHAPD
3. Kth Max Subarray: https://www.codechef.com/problems/KTHMAX
4. Across the River: https://www.codechef.com/problems/RIVPILE

2.2 Problem set JA

Target date: **May 5**

1. Lighthouses: https://www.codechef.com/problems/LIGHTHSE
2. Xor-tree: http://codeforces.com/contest/429/problem/A
3. Product of Diameters: https://www.codechef.com/problems/TREEDIAM
4. Range Minimum Queries: https://www.codechef.com/problems/ANDMIN

2.3 Problem set KA

Target date: **May 6**

1. Oracle Devu and Longest Common Subsequence: https://www.codechef.com/problems/ORACLCS
2. BerSU Ball: http://codeforces.com/contest/489/problem/B
3. Yet Another SubSegment Sum Problem: https://www.codechef.com/problems/SEGSUMQ
4. Möbius function and intervals:[1] https://projecteuler.net/problem=464

[1] Don't worry, this is more algorithmic than mathematical! We've covered everything you need.

2.4 Problem set KS

Target date: **May 7**

1. Nice SubSegments: https://www.codechef.com/problems/SUBSGM
2. Sebi and the corrupt goverment:[2] https://www.codechef.com/problems/SETELE
3. Guessing Game: https://projecteuler.net/problem=406
4. Island Puzzle: http://codeforces.com/problemset/problem/627/F

2.5 Problem set PY

Target date: **May 8**

1. Uphill paths: https://projecteuler.net/problem=411
2. Chef and Reversing: https://www.codechef.com/problems/REVERSE
3. Time Travelling Monster: https://www.codechef.com/problems/TIMETRAV
4. Merciless Chef: https://www.codechef.com/problems/MLCHEF

2.6 Problem set TD

Target date: **May 9**

1. Superinteger: https://projecteuler.net/problem=467
2. Niceness of a tree: https://www.codechef.com/problems/NICENESS
3. Binary Circles: https://projecteuler.net/problem=265
4. Perfect Subarrays: https://www.codechef.com/problems/SUBARR

[2] "Expected value", in this case, is simply the average. For example, the expected value of throwing a dice is the *average across all events*, i.e., $\frac{1+2+3+4+5+6}{6} = 3.5$. This works because all events are equally likely; in other cases, you should weigh the events by their probability.

2.7 Problem set TR

Target date: **May 10**

1. Forest Gathering: https://www.codechef.com/problems/FORESTGA
2. New Year Permutation: http://codeforces.com/contest/500/problem/B
3. Almost Union-Find: https://uva.onlinejudge.org/external/119/11987.pdf (UVa 11987)
4. Line Intersections: https://www.codechef.com/problems/CHN15D

2.8 Problem set VG

Target date: **May 11**

1. Roses for Alexey: https://www.codechef.com/problems/ALEXROSE
2. Parity tree: https://www.codechef.com/problems/PARITREE
3. Devu and Manhattan Distance: https://www.codechef.com/problems/MDIST
4. Tree: https://www.codechef.com/problems/RRTREE

2.9 Notes

For Project Euler problems, consider your code accepted if it can *compute* the answer in less than 10 seconds, and by that, I mean actually computing the answer, not just hardcoding $+$ printing it (or something of similar “cheaty” flavor).

For problems with subtasks, partial solutions will get partial points, although you'll only get the extra 1 point per set if you get full points for all 4.