The Ultimate Guide for the ICPC

Resources to prepare yourself for the The International Collegiate Programming Contest (ICPC) or any competitive programming contest

Hassan El Desouky
Sep 19, 2019 · 3 min read ★

Photo by Ariel Besagar on Unsplash

What is the ICPC?

> *"The ICPC, the 'International Collegiate Programming Contest,' is an extra-curricular, competitive programming sport for students at universities around the world. ICPC competitions provide gifted students opportunities to interact, demonstrate, and improve their teamwork, programming, and problem-solving process. The ICPC is a global platform for academia, industry, and community to shine the spotlight on and raise the aspirations of the next generation of computing professionals as they pursue excellence." — Wikipedia*

Disclaimer

I'm by no mean a top coder or a world finalist. I just wanted to share the resources I found helpful, and if you follow them, hopefully, you will be an ICPC world finalist.

. . .

Introduction

I really like competitive programming, and I really want to be someone like

I really like competitive programming, and I really want to be someone like Gennady Korotkevich. He is truly a living legend in the field of competitive programming.

Anyways, let’s get into the details …

The Steps to be a Good Competitive Programmer

It’s really simple: There are no tricks and there’s no short way … all you need is dedication and a goal, and you are all set.

First, you will need to be familiar with at least one of the following programming languages: Python, Java, C, or C++.

Secondly, you will need to study all of the different topics about data structures and algorithms.

Finally, do a lot of problems … *a lot!*

Topics

These are the main topics that should be done thoroughly.

Number theory

1. Euclidian and extended Euclidian algorithm
2. Modular arithmetic and modular inverse
3. Prime generation (sieve and segmented sieve)
4. Fermat’s theorem
5. Euler’s Totient function
6. Miller Rabin primality test
7. Chinese remainder theorem
8. Lucas theorem

Greedy algorithms

1. Activity-selection problem
2. Kruskal’s algorithm
3. Prim’s algorithm

Binary search

1. Topcoder binary search
2. Binary search
3. Ubiquitous binary search — get a grasp of discrete and continuous binary searches

Data structures

1. Linked lists
2. Binary-search tree
3. Binary-indexed tree or Fenwick tree
4. Segment Tree (RMQ, range sum, and lazy propagation)
5. Red-Black trees
6. Hashing
7. Extensive list of data structures

Graph algorithms

1. Breadth-first search (BFS)
2. Depth-first search (DFS)
3. Shortest path from source to all vertices (Dijkstra)
4. Shortest path from every vertex to every other vertex (Floyd Warshall)
5. Minimum spanning tree (Prim)
6. Minimum spanning tree (Kruskal)
7. Topological Sort
8. Johnson's algorithm
9. Articulation points (or cut vertices) in a graph
10. Bridges in a graph
11. All graph algorithms

String algorithms

Learning library functions for string actually proves very helpful. (C++: See this, this, String in Java.)

1. KMP algorithm
2. Rabin karp
3. Z's algorithm
4. Aho-Corasick string matching
5. Suffix arrays
6. Trie
7. Finite automata

Dynamic programming

1. Dynamic programming — GeeksforGeeks
2. Dynamic Programming — Codechef

Dynamic programming is quite important and can be infused and asked with various other topics. Some different types of DP concepts are:

Classic DP

1. Longest-common subsequence
2. Longest-increasing subsequence
3. Edit distance
4. Minimum partition
5. Ways to cover a distance
6. Longest path in matrix
7. Subset-sum problem
8. Optimal strategy for a game
9. 0–1 knapsack problem
10. Assembly-line scheduling
11. All DP algorithms

Computational geometry

1. Convex-hull algorithms
2. Geometric algorithms

. . .

Resources

“Competitive Programming 3”: This book is beyond great. I’m currently in the middle of it and I’m enjoying the problems and the book structure. I highly recommended it.

“The ‘Science’ of Training in Competitive Programming”: In this blog,the author explains how to train for CP and what he did in order to be a good problem solver.

How to be a red coder: This Quora answer really goes deep into how to be a red coder and train efficiently.

Programming | Software Engineering | Coding | Competitive Programming | Guides And Tutorials

282 claps

WRITTEN BY

Hassan El Desouky

CS Student Developer Optimist "What you do at 8pm will determine your future."

Follow

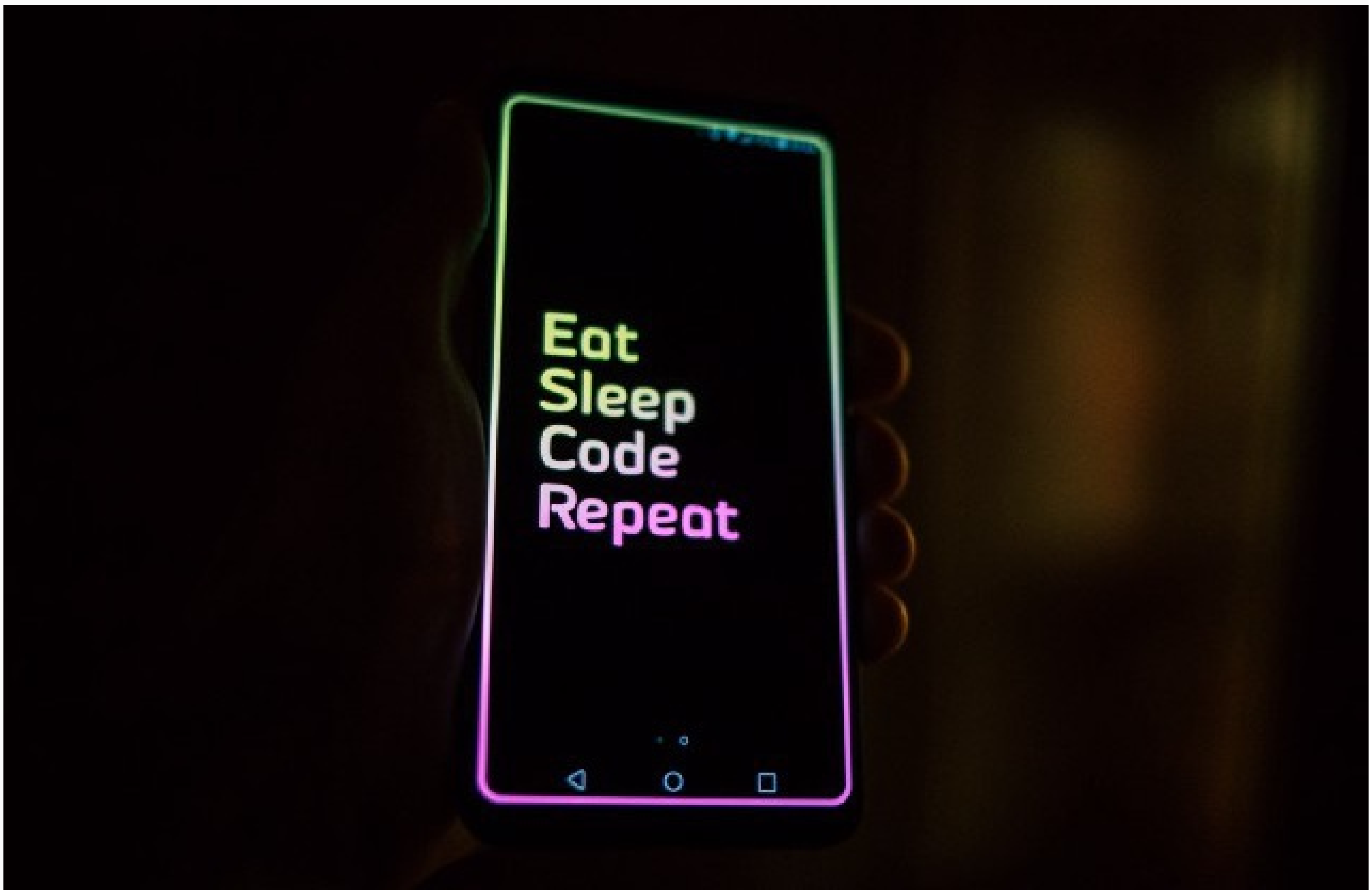
Better Programming

Advice for programmers.

Follow

More From Medium

More from Better Programming



9 Things to Know to Master List Comprehensions in Python

Yong Cui, Ph.D. in Better Programming
Mar 3 · 4 min read ★

1.6K

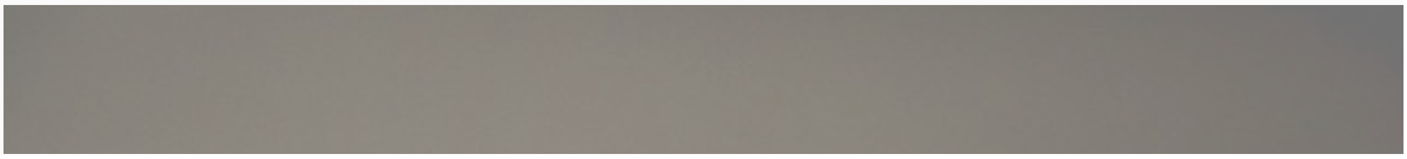
More from Better Programming

6 Must-Use Tools for Front-End Development

Mahdhi Rezvi in Better Programming
Mar 4 · 4 min read ★

2.5K

More from Better Programming



Dictionary Merging and Updating in Python 3.9

Yong Cui, Ph.D. in Better Programming
Mar 6 · 5 min read ★

847