Stanford Alumni Association

Technical Interviewing

for Entry Level Software Engineering Roles

Yongwhan Lim

Wednesday, October 11, 2023

Yongwhan Lim

Education

Stanford University

MIT

Part-time Jobs

Cornell Tech

MIT EECS

Full-time Job

Google Research

TWO SIGMA

Workshops

Stanford ENGINEERING | Stanford Computer Forum

Carnegie Mellon University

Harvard

University of California

KAIST

NUS National University of Singapore

Coach/Judge

icpc International Collegiate Programming Contest

icpc.foundation

advancing the art and sport of competitive programming

https://www.yongwhan.io

Yongwhan Lim

- Currently:
  - **CEO** (Co-Founder) in a Stealth Mode Startup;
  - **Co-Founder** in Christian and Grace Consulting;
  - ICPC **Internship Manager**;
  - ICPC **North America Leadership** Team;
  - Columbia ICPC **Head Coach**;
  - ICPC **Judge** for NAQ and Regionals;
  - **Adjunct** (Associate in CS) at Columbia;
  - **Visiting Instructor** at Cornell-Tech;

https://www.yongwhan.io

NOW... IT IS ABOUT YOU!

- Please fill out a survey on: **https://bit.ly/stanford-workshop-survey**

NOW!

- I will give you **few minutes** to fill out the survey :)

Overview

- **Part I: Interview Preparation**
  - Interview Types
  - Technical Interview
  - Interview Topics
  - Interview Preparation Resources
- **Part II: Competitive Programming**
  - CodeForces
  - ICPC

Part I: Interview Preparation

Interview Types

- Technical Interview
  - Tests technical skill-sets required for a job.
- Behavioral Interview
  - Tests soft skills (e.g., effective communication, conflict resolution, etc)

Interview Types

- **Technical Interview**
  - Tests technical skill-sets required for a job.
- Behavioral Interview
  - Tests soft skills (e.g., effective communication, conflict resolution, etc)

Technical Interview

- Recruiter Call
- 0-1 Online Coding Challenge
  - automated screening with 2-3 questions.
- 2-3 Technical Phone Screens
  - first technical conversation with human.
- 4-7 Interviews in Onsite
  - similar to phone screening but more in-depth; you may get probed on your claimed expertise.
- 0-5 Fit Calls & Negotiation

Technical Interview

- Recruiter Call
- **0-1 Online Coding Challenge**
  - automated screening with 2-3 questions.
- **2-3 Technical Phone Screens**
  - first technical conversation with human.
- **4-7 Interviews in Onsite**
  - similar to phone screening but more in-depth; you may get probed on your claimed expertise.
- 0-5 Fit Calls & Negotiation

Interview Topics

- Data Structures and Algorithms
- (> entry level) System Design Problems

Interview Topics

- **Data Structures and Algorithms**
- (> entry level) System Design Problems

Interview Topics

- **Fundamentals**
  - Arrays and Linked Lists
  - Binary Trees
  - Heaps
  - Sorting

Interview Topics

- **Important**
  - Stacks and Queues
  - Hash Tables
  - Binary Search Trees
  - Searching
  - Recursion

Interview Topics

- **Real Differentiators**
  - **Strings**: Knuth Morris Pratt (KMP); Rabin Karp / String Hashing; Suffix Array; Suffix Automaton;
  - **Dynamic Programming**: 1D; 2D; Interval; Tree;
  - **Greedy Algorithms** and Invariants: Matroid;
  - **Graphs**: Shortest Path; LCA (Lowest Common Ancestor); Flow (Ford Fulkerson & Edmond Karp) / Matching; Minimum Spanning Tree (Prim / Kruskal);
    - **BFS** (Breadth First Search); **DFS** (Depth First Search): unweighted
    - **Dijkstra** (Heap/Priority Queue): weighted (non-negative)
    - **Bellman-Ford**: weighted (can have negative edges)
    - **Floyd Warshall**: all-pair shortest path

Sample Problem on String

- A minimum number of insertions to make a string a palindrome.

Sample Problem on String

- A minimum number of insertions to make a string a palindrome.

- **Constraint**
  - string length is at most 5000
  - each character is from ‘a’ to ‘z’

Sample Problem on String

- A minimum number of insertions to make a string a palindrome.

- **Constraint**
  - string length is at most 5000
  - each character is from ‘a’ to ‘z’

Any Idea?

Model Solution

```
int minInsertions(string &s) {
    int n = s.size();
    vector<vector<int>> dp(n, vector<int>(n,0));
    for (int i = 1; i < n; i++)
        for (int j = 0, k = i; k < n; j++, k++)
            dp[j][k] = (s[j]==s[k]) ?
                            dp[j+1][k-1] :
                            min(dp[j][k-1],dp[j+1][k])+1;
    return dp[0][n-1];
}
```

Interview Preparation Resources

- **Popular Websites**
  - LeetCode: Solve all four weekly/biweekly problems in **20 minutes**!
    - 1+2+4+8 (+5 buffer)
  - CodeForces: Get to 2200+ rating
    - Clear 5 questions out of 6 **fast**!
  - AtCoder; TopCoder; CodeChef;

- **Annual Contests**
  - Meta Hacker Cup; ~~Google Code Jam; TopCoder Open;~~

Interview Preparation Resources

- Elements of Programming Interview
- Competitive Programming 4

Part II: Competitive Programming

CodeForces

- Get to **2200+** rating as fast as you can!

- Join the training sessions through **Programming Zealots** (**https://bit.ly/programming-zealot**).

Success Pathways

- Those who are just starting should focus on the **first half** of problems in Zealot Problem Set. Your main focus should be gaining some experiences with an explicit goal to enjoy the process of solving new problems and potentially making it to the ICPC North America Championship (NAC)!

- Those who are more serious should focus on the **second half** of problems in Zealot Problem Set. Your goal should be making into the World Finals and potentially winning a medal!

Practice Strategies

- If your goal is to get to a rating of **X**, you should practice on problems that are **X + 300** typically, with a spread of 100. So, picking problems within the range of:

  {X + 200, X + 300, X + 400}

  would be sensible!

- So, if you want to target becoming a **red**, which has a lower-bound of 2400, you should aim to solving {2600, 2700, 2800}.
- **(Eventual) Target**: You should focus on solving it for 30 minutes or less!

Practice Strategies

- You should focus on solving each problem for **30 minutes or less**; if you cannot solve any problem with this range, you should consider solving a problem with a lower rating.

- You should aim to solve **10 ~ 15 problems** each day within this range to expect a rank up within a quarter (3 months).

Practice Strategies

- If you cannot solve a problem, here is a sample recipe you can follow:

Practice Strategies

- If you cannot solve a problem, here is a sample recipe you can follow:
  - Look at editorial for hints, and try to solve the problem.

Practice Strategies

- If you cannot solve a problem, here is a sample recipe you can follow:
  - Look at editorial for hints, and try to solve the problem.
  - Look at editorial for full solutions, and try to solve the problem.

Practice Strategies

- If you cannot solve a problem, here is a sample recipe you can follow:
  - Look at editorial for hints, and try to solve the problem.
  - Look at editorial for full solutions, and try to solve the problem.
  - Look at accepted solutions, and try to solve the problem.

Practice Strategies

- If you cannot solve a problem, here is a sample recipe you can follow:
  - Look at editorial for hints, and try to solve the problem.
  - Look at editorial for full solutions, and try to solve the problem.
  - Look at accepted solutions, and try to solve the problem.
  - Make sure you look back after two weeks and see if you can solve it.

International Collegiate Programming Contest (ICPC)

- If you would like to get involved in helping out as a volunteer, please reach out to me at **yongwoods@icpc.global**.

Workshop Feedback Survey

- Please complete the following survey: **https://bit.ly/stanford-workshop-feedback**

Contact Information

- **Email**: yongwhan@yongwhan.io

- **Personal Website**: https://www.yongwhan.io/

- **LinkedIn Profile**: https://www.linkedin.com/in/yongwhan/
  - Feel free to send me a connection request!
  - Always happy to make connections with promising students!