

Amazon ও Google চাকরির সুযোগ পাওয়ার প্রস্তুতি পর্ব-১

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 1 MINUTE

ভূমিকা:

প্রথমেই আমার লেখায় বাংলা-ইংরেজী ভয়াবহ সংমিশ্রনের জন্য দুঃখিত। আমার ইচ্ছা ছিল পুরোটা বাংলায় লিখব, কিন্তু লিখতে গিয়ে মনে হয়েছে কিছু শব্দ ইংরেজিতে রাখাই ভাল।

এই লেখার উদ্দেশ্য পুরো প্রস্তুতির সব কিছু এক সাথে রাখা, কারন আমি যখন প্রস্তুতি নেয়া শুরু করি তখন বিভিন্ন জায়গা থেকে তথ্য নিতে হয়েছে, তখন মনে হয়েছে যদি সব এক সাথে থাকত।

আমি অন্য যাদের লেখা পড়েছি সবার নিজস্ব ধরন ছিল প্রস্তুতির। যেমন একজন লিখছে যেহেতু ইন্টারভিউ একটানা ৩/৪ ঘন্টা হয়, তাই প্রস্তুতি সময় একটানা পড়তে, কিন্তু আমি ২০/৩০ মিনিট পড়েই ছোট বিরতি নিতাম। আর একটা লেখায় পড়েছিলাম, LeetCode এ ৭০ টা ভালো প্রবলেম সমাধান করে যদি কেউ ভালো ইন্টারভিউর প্রস্তুতি নিতে না পারে, তাহলে ৭০০ টা সমাধান করেও ভালো ইন্টারভিউ দিতে পারবে না। কিন্তু আমার কাছে মনে হয়েছে যেহেতু আমি অনেক দিন Competitive Programming করি নাই, তাই আমি ৭০ টা ভালো প্রবলেম করে পারব না, কিন্তু ৭০০ টা করলে অভিজ্ঞতার জন্য By Instinct অনেক কিছু মাথায় আসবে। তবে এমন না যে আমার ধরনটাই সবার জন্য প্রযোজ্য হবে, শুধুমাত্র একটা ধারণা দেয়ার জন্যই লিখা।

আমার প্রস্তুতি সময় ছিল ৬ মাস। আমার মোটামুটি বেসিক সব Data Structure and Algorithm নিয়ে ধারণা থাকায় কোন বই না পরে সরাসরি Leetcode এ অনুশীলন করা শুরু করি। কারো যদি Data Structure and Algorithm এ কোন দুর্বলতা থাকে তাহলে আগে ঐগুলো ভালো ভাবে বই বা অন্য রিসোর্স থেকে পড়ে বুঝতে হবে।

নিচের অংশ পড়ার আগে

https://www.youtube.com/channel/UC4xKdmAXFh4ACyhpiQ_3qBw এ গিয়ে

একটা টুঁ মেরে আসার অনুরোধ থাকল। বিশেষ করে নিচের লিঙ্কগুলো।

[How to learn to code \(quickly and easily!\)](#),

[Why I left my job at Google \(as a software engineer\)](#),

[my wife left me. \("How success destroyed us"\)](#)

প্রথমে ইন্টারভিউর সব ধাপ সম্পর্কে বলি।

ইন্টারভিউর ধাপ:

Resume:

ভালো Resume তৈরী করা চাকরি খোঁজার প্রথম কাজ। এখন Resume কয় পাতা হবে, আমার Recommendation থাকবে ১ পাতা। আমি প্রথমে ২ পাতার Resume তৈরী করেছিলাম, পরে ১ পাতায় নিয়ে আসি। Google, Amazon এ এই ১ পাতার Resume দিয়েই ইন্টারভিউ কল পেয়েছি। আমি এই সাইট থেকে <https://app.zety.com/user/cv> template ব্যবহার করেছি। Resume তে বিভিন্ন Programming Skills রেটিং আকারে না দিয়ে Expert, Familiar, Advanced এইভাবে দিলে ভালো। No Typo, No Buzz Word। Resume তে বিভিন্ন প্রজেক্ট এর সাথে GitHub লিংকও দেয়া উচিত। বিশেষত যদি, Mid-Range Software Company তে চাকরির জন্য আবেদন করা হয়। Resume তে ছবি, জন্ম তারিখ, রেফারেন্স অবশ্যই দেয়া যাবে না। Recruiter রা সাধারণত গড়ে ৬ সেকেন্ড সময় নিয়ে Resume দেখেন

<https://medium.com/free-code-camp/writing-a-killer-software-engineering-resu>

[me-b11c91ef699d](#), কাজেই এই ৬ সেকেন্ডে যাতে নিজের সম্পর্কে আগ্রহী করে তোলা যায়, এটা মাথায় রেখে Resume তৈরি করতে হবে।

ইন্টারভিউ কল:

দ্বিতীয় কাজ হচ্ছে ইন্টারভিউর কল আসার ব্যবস্থা করা। এক্ষেত্রে LinkedIn এ সরাসরি চাকরিতে ইন্টারভিউর জন্য আবেদন করা যায়। LinkedIn এ Premium Account ও একটা ভালো উপায়, বিভিন্ন কোম্পানির Hiring Manager / Recruiter কে সরাসরি মেসেজ [InMail] দেয়া যায়। সবচেয়ে ভালো উপায় হল referral ব্যবহার করা। কোন কোম্পানিতে কাজ করে এমন কাউকে দিয়ে চাকরির জন্য আবেদন করানো। Referral এর জন্য খুব ভালো ভাবে পরিচিত লাগবে, এইটা দরকারী না, Undergrad/Grad School এ পরিচিত বা সিনিয়র/জুনিয়র কিছু একটা হলেই হবে। আমি LinkedIn এ খুঁজে খুঁজে Stony Brook University বা Bangladesh বা SUST এর কেউ হলে LinkedIn এ মেসেজ দিতাম। অনেকেই হয়ত ব্যস্ততার কারণে উত্তর নাও দিতে পারে, অথবা Referral এর সুযোগ নাও থাকতে পারে, এটাও মাথায় রাখতে হবে।

ইন্টারভিউ Recruiter:

প্রথমে সাধারণত Recruiter রা ফোন দেয়। ওরা মূলত বর্তমান পড়ালেখা বা চাকরির অবস্থা, কোন বিষয়ে কাজ করতে আগ্রহী, কবে ইন্টারভিউ দিতে পারবে এই সব কথা বলে। এখানে Recruiter রাই ইন্টারভিউ থেকে শুরু করে Job Offer, Salary Negotiation সব কিছু করে থাকে। ইন্টারভিউ কবে দিব, ফলাফল কি, Onsite ইন্টারভিউ হলে আসা-যাওয়া, থাকা-খাওয়া মোটামুটি সবই Recruiter রাই ব্যবস্থা করে দেয়। ইন্টারভিউতে প্রশ্ন কেমন হতে পারে, কি কি বিষয়ে পড়তে হবে, গুরুত্বপূর্ণ অনেক Resource ও দিয়ে থাকে। কোন বিষয়ে প্রশ্ন থাকলে Recruiter কে বলে জেনে নেয়া ভালো। যেমন আমাকে একটা ইন্টারভিউতে প্রথমে Machine Learning Engineer পোস্টের জন্য Recruiter

আবেদন করতে বলেছিল, কিন্তু আমি কথা বলে পরিবর্তন করে Software Engineer পোস্টের জন্য ব্যবস্থা করতে বলি।

Online Round:

অনেক কোম্পানি প্রথমে Online Round নেয়। এইটা Recruiter মেইলে লিংক দিয়ে ৭/১০ দিন সময়ের ভিতরে করতে বলে। সাধারণত ১/২ ঘন্টা সময়ে ২/৫ টা প্রবলেম সমাধান করতে হয়। অনেক সময় সাথে কিছু MCQ প্রশ্নও থাকে।

Phone Screen:

সাধারণত ২/৩ টা Phone Screen ইন্টারভিউ হয়। Phone Screen ইন্টারভিউতে Google doc বা <https://codeshare.io/> তে Code করতে হয়। Interviewer Skype, Hangout বা Phone এ কথা বলে। প্রথমে প্রবলেম কিভাবে সমাধান করব, উদাহরণ দিয়ে বুঝাতে হয়, Time and Space Complexity নিয়ে আলোচনা করতে হয়, যদি Interviewer সবুজ সংকেত দেয়, তবেই Code লেখা শুরু করা উচিত। Phone screen ৪৫ মিনিট থেকে ১ ঘন্টা হয়। প্রথম ৫-১০ মিনিট নিজের সম্পর্কে বলতে হয়, Interviewer ও নিজের কথা বলে। এর পরে ৩০-৪৫ মিনিট ২ টা প্রবলেম সমাধান করতে হয়। কখনও দুইটা ভিন্ন প্রবলেম থাকে, কখনও প্রথম প্রবলেমের Time and Space Complexity Optimize করতে হয়।

Onsite:

আগের ধাপ Successfully শেষ হলে Onsite এ ইন্টারভিউ। Onsite এ ৪/৫ রাউন্ড থাকে। প্রত্যেকটা ৪৫ মিনিট থেকে ১ ঘন্টা, কোন Break ছাড়া, তবে Lunch Break থাকতে পারে। Onsite ইন্টারভিউগুলোতে Phone Screen এর মত প্রশ্ন থাকে। তবে Code করতে হয় Whiteboard এ।

যদিও ইদানীং Google এ Chromebook দেওয়া হয়। আমার কাছে Whiteboard ই ভালো, নিজের চিন্তা ভাল ভাবে ব্যাখ্যা করে যায়, Code এ যদি কোন পরিবর্তন থাকে, তাহলে সহজে করা যায়।

এই মোটামুটি ইন্টারভিউ প্রক্রিয়া। (চলবে)

Amazon ও Google চাকরির সুযোগ পাওয়ার প্রস্তুতি পর্ব-২

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 1 MINUTE
প্রস্তুতি Resource:

১। Cracking the Coding Interview <http://www.crackingthecodinginterview.com/>
দিয়ে আমার প্রস্তুতির শুরু। এই বইয়ে ইন্টারভিউর সব কিছুই একটা ধারণা পাওয়া যায়। যদিও এই বইয়ের প্রশ্ন খুব কমন, তাই এগুলো ইন্টারভিউতে আসেনা, তবে ইন্টারভিউতে আসলে কেউ যদি না পারে, তাহলে ধরে নেয়া, Homework করা হয় নাই। MUST READ BOOK.

২। Elements of Programming Interview,
<https://elementsofprogramminginterviews.com/>

৩। LeetCode Premium Account <https://leetcode.com/problemset/all/> : Premium Account এর সুবিধা হল এইটায় কোম্পানি tag থাকে, বড় কোম্পানির আলাদা section আছে, Mock ইন্টারভিউ দেয়া যায়। অন্তত পক্ষে ২ মাসের জন্য হলেও LeetCode Premium Account নিয়ে Code করা উচিত। USA তে চাকরি পেতে হলে LeetCode MUST.

৪। Geeksforgeeks <https://www.geeksforgeeks.org/>: এই সাইটে মূলত আমি বিভিন্ন প্রশ্নের
ভিন্ন ভিন্ন উত্তর জানার জন্য দেখেছি।

৫। Back to Back SWE

<https://www.youtube.com/channel/UCmJz2DV1a3yfgR7GqRtUUA>: এই চ্যানেল
LeetCode এর বিভিন্ন সমাধান এর ব্যাখ্যা দেয়া আছে। আমার কাছে খুব ভালো লেগেছে।

৬। Algorithm Every Day

<https://www.youtube.com/channel/UCx-kFfzekMbHODaBss-ZnsA>: এই চ্যানেল
LeetCode এর বিভিন্ন সমাধান এর ব্যাখ্যা দেয়া আছে।

৭। Grokking The System Design Interview

<https://www.educative.io/collection/5668639101419520/5649050225344512>:
System Design এর জন্য এইটা MUST READ। যদিও paid course, কিন্তু System Design
এর জন্য এই কোর্স আমার কাছে খুবই ভালো লেগেছে।

৮। System Design

<https://www.youtube.com/channel/UCRPMaQdtSgdoIpeef7iFsKw>: এই চ্যানেল
System Design এর জন্য MUST।

৯। Dan Crator

<https://www.youtube.com/channel/UCwouQHve23oMWgQcTTpgQsQ>: এই চ্যানেল
যদিও Amazon specific Behavioral Question, তবে যে কোন Company তে প্রযোজ্য হবে।

১০। <https://www.pramp.com/dashboard#/>, ইন্টারভিউ প্র্যাকটিস এর জন্য MUST. আমি প্রায় ৩০টার মত ইন্টারভিউ দিয়েছি এইখানে। এই সাইটে অন্য যারা ইন্টারভিউ দিবে তারা আমার ইন্টারভিউ নেয়, পরে আমি ওদের। প্রশ্ন সাইট থেকে দেয়া হয়, সাথে সমাধান এবং hints ও দেয়া থাকে।

১১। <https://github.com/donnemartin/system-design-primer>, এই লিংকেও System Design এর ভালো Content আছে।

১২।

<https://medium.com/@scarletinked/are-you-the-leader-were-looking-for-interviewing-at-amazon-8301d787815d>, Amazon LP প্রশ্নের জন্য খুবই ভালো। (চলবে)

পর্ব-১

<https://www.facebook.com/notes/sabir-ismail/amazon-%E0%A6%93-google-%E0%A6%8F-%E0%A6%9A%E0%A6%BE%E0%A6%95%E0%A6%B0%E0%A6%BF%E0%A6%B0-%E0%A6%AA%E0%A7%8D%E0%A6%B0%E0%A6%B8%E0%A7%8D%E0%A6%A4%E0%A7%81%E0%A6%A4%E0%A6%BF-%E0%A6%AA%E0%A6%B0%E0%A7%8D%E0%A6%AC-%E0%A7%A7/10157715186369674/>

Amazon ও Google চাকরির সুযোগ

পাওয়ার প্রস্তুতি পর্ব-৩

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 2 MINUTES

সময় বন্টন:

১ম মাস: LeetCode Easy Problem

প্রথমে আমি Cracking The Coding Interview দিয়ে পড়া শুরু করি। এই বইয়ের Data Structures and Algorithm সম্পর্কিত সব চ্যাপ্টার প্রথমে শেষ করি। এতে বেসিক আইডিয়াগুলো Revised হয়। প্রথমবার এই বইয়ের প্রবলেম সমাধান করার সময় Naïve Approach এ করেছি। যেই প্রবলেম সমাধান করতে পারি নাই, ঐ গুলো আপাতত বাদ দিয়ে যাই। এরপর LeetCode করা শুরু। Research এর জন্য Code করা হলেও আমি অনেক দিন থেকে Data Structures and Algorithm সম্পর্কিত Code করা বন্ধ ছিল। তাই প্রথম মাসে টার্গেট ছিল LeetCode এ শুধু Easy প্রবলেম, প্রতিদিন ১০ টা করে, ৩০০ টা করব। LeetCode এ প্রবলেমগুলো খুব ছোট থাকে।

Easy:

<https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array/>

[https://leetcode.com/problems/single-number,](https://leetcode.com/problems/single-number/)

[https://leetcode.com/problems/single-number-ii/,](https://leetcode.com/problems/single-number-ii/)

<https://leetcode.com/problems/single-number-iii/>

<https://leetcode.com/problems/two-sum/>

<https://leetcode.com/problems/longest-common-prefix/>

<https://leetcode.com/problems/majority-element/>

আমি একটা প্রবলেম এ ১৫/২০ মিনিটের বেশী সময় দিতাম না। এর মধ্যে না হলে সমাধান দেখে ফেলতাম। এছাড়া Stack, Queue, HashMap, Linked List, Binary Tree, Set, Disjoint Set, Tries, বিভিন্ন ধরনের Sorting, Binary Search, Heap এইগুলার Scratch থেকে Implementation + Python এর Library দিয়ে ব্যবহার করলে এইগুলার সব ধরনের Time Complexity যেমন Insert, Delete, Search

<https://wiki.python.org/moin/TimeComplexity> এইগুলো এই মাসে দেখেছি। Interviewer রা Code Production Ready দেখতে পছন্দ করেন, আমার যেহেতু Industry Experience নাই, তাই এই লিংক থেকে <https://pep8.org/> Python এর Coding Standard Review দিয়েছি।

Time and Space Complexity:

যে কোন প্রবলেমের Time and Space Complexity Analysis অবশ্যই করতে হবে। এইটা ভালো করে শিখতে হবে, বিশেষ করে Recursive প্রবলেমের Time and Space Complexity Analysis। খুব Accurate না হলেও কাছাকাছি যাতে বের করা যায়। LeetCode এ মোটামুটি সব প্রবলেমের Time and Space Complexity Analysis Discussion এ থাকে।

২য় মাসঃ Data Structure: Linked List, Binary Tree, HashMap, Heap

Algorithm থেকে Data Structure বেশী গুরুত্বপূর্ণ। Algorithm এ সর্বোচ্চ DFS / BFS আসে। Dijkstra, Bellman Ford বা Prims Algorithm, এইগুলো কপাল খুব খুব খারাপ না হলে আসার কথা না। এই মাসে LeetCode এ Linked List, Binary Tree, HashMap, Heap এর প্রায় সব প্রবলেম Hard ছাড়া সমাধান করেছে। LeetCode ছাড়াও Geeksforgeeks থেকেও আরও LinkedList, Binary Tree এর প্রবলেম সমাধান করেছে। LinkedList, Binary Tree দিয়ে মোটামুটি ২০০ বেশী এর প্রবলেম সমাধান করেছে। মনে হতে পারে, Binary Tree দিয়ে কি এত প্রবলেম আছে? Binary Tree Left View, Right View, Top View, Bottom View, Boundary View, Depth Order, Level Order, Vertical Order, Zigzag Order, Inorder, Preorder, Postorder শুধু Tree Traversal দিয়েই ১১ টা প্রশ্ন।

Binary Tree:

<https://leetcode.com/problems/validate-binary-search-tree/>

<https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree/>

<https://leetcode.com/problems/boundary-of-binary-tree/>

<https://leetcode.com/problems/verify-preorder-sequence-in-binary-search-tree/>

<https://leetcode.com/problems/binary-tree-maximum-path-sum/>

Linked List:

<https://leetcode.com/problems/merge-two-sorted-lists/>

<https://leetcode.com/problems/linked-list-cycle/>

<https://leetcode.com/problems/swap-nodes-in-pairs/>

<https://leetcode.com/problems/reverse-nodes-in-k-group/>

<https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list/>

Heap:

কোন প্রবলেম এ k-most টার্ম থাকলেই ধরে নিতে হবে এইটা সম্ভবত Heap এর প্রবলেম। এছাড়াও যদি কোন প্রবলেম $O(n)$ complexity তে সমাধান করার পর Interviewer আরো Optimization করতে বলে বা logarithm Complexity তে সমাধান চায়, তাহলেও more likely এটা Heap দিয়ে সমাধান করা যাবে।

<https://leetcode.com/problems/merge-k-sorted-lists/>

<https://leetcode.com/problems/find-median-from-data-stream/>

<https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/>

<https://leetcode.com/problems/k-closest-points-to-origin/>

<https://leetcode.com/problems/path-with-maximum-minimum-value/>

HashMap:

Python এ set, dict, Counter, defaultdict, OrderedDict এইগুলোর Basic ভালো করা জানতে হবে, কোনটার সাথে কি পার্থক্য, Search(), Insert(), Delete() এর Time and Space Complexity।

<https://leetcode.com/problems/design-hashmap/>

<https://leetcode.com/problems/design-hashset/>

<https://leetcode.com/problems/max-points-on-a-line/>

<https://leetcode.com/problems/number-of-distinct-islands/>

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

পর্ব-২

<https://www.facebook.com/notes/sabir-ismail/amazon-google-%E0%A6%9A%E0%A6%BE%E0%A6%95%E0%A6%B0%E0%A6%BF%E0%A6%Bo-%E0%A6%AA%E0%A7%8D%E0%A6%B0%E0%A6%B8%E0%A7%8D%E0%A6%A4%E0%A7%81%E0%A6%A4%E0%A6%BF-%E0%A6%AA%E0%A6%B0%E0%A7%8D%E0%A6%AC-%E0%A7%A8/10157715195489674/>

Amazon ও Google চাকরির সুযোগ পাওয়ার প্রস্তুতি পর্ব-৪

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 3 MINUTES

৩য় মাস: More Data Structure: Stack, Two Pointer, Sliding Window, Binary Search, Tries, DFS, BFS

এই মাসেও Data Structure প্রবলেম সমাধান করেছি। Stack দিয়ে অনেক Hard প্রবলেম সমাধান করা যায়। যেমন:

Stack:

<https://leetcode.com/problems/largest-rectangle-in-histogram/>

<https://leetcode.com/problems/remove-duplicate-letters/>

<https://leetcode.com/problems/next-greater-element-i/>

<https://leetcode.com/problems/next-greater-element-ii/>

<https://leetcode.com/problems/daily-temperatures/>

Next largest, smallest বা parenthesis সম্পর্কিত কোন প্রবলেম আসলে প্রথমেই Stack মাথায় আসত। যদিও প্রথম প্রথম এই প্যাটার্ন প্রবলেম ধরতে পারাটা কঠিন। এর জন্য প্রবলেম সমাধান করে অন্যদের সমাধান দেখাও দরকার, যাতে আইডিয়া পাওয়া যায়।

Two Pointer and Sliding Window:

এই দুইটা আগে জানা ছিল না, এই দুইটাও খুব কাজের জিনিস। কোন প্রবলেমে sub-string নিয়ে কাজ করা লাগলে সাধারণত Two Pointer বা Sliding Window মাথায় আসত।

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

<https://leetcode.com/problems/3sum/>

<https://leetcode.com/problems/trapping-rain-water/>

<https://leetcode.com/problems/longest-substring-with-at-most-two-distinct-characters/>

<https://leetcode.com/problems/sliding-window-maximum/>

Binary Search:

Binary Search দিয়েও অনেক Hard প্রবলেম সমাধান করা যায়। এখানেও প্রথম প্রথম প্যাটার্নটা ধরতে পারা কঠিন।

<https://leetcode.com/problems/split-array-largest-sum/>

<https://leetcode.com/problems/median-of-two-sorted-arrays/>

<https://leetcode.com/problems/russian-doll-envelopes/>

<https://leetcode.com/problems/count-of-smaller-numbers-after-self/>

<https://leetcode.com/problems/search-in-rotated-sorted-array/>

Tries:

Tries মূলত Time Complexity Optimize কাজে লাগে। DFS / BFS অনেক প্রবলেমের Naïve Solution হয়ত আছে, কিন্তু ইন্টারভিউতে Tries দিয়ে সমাধান করতে হয়।

<https://leetcode.com/problems/implement-magic-dictionary/>

<https://leetcode.com/problems/word-search-ii/>

<https://leetcode.com/problems/word-squares/>

<https://leetcode.com/problems/longest-word-in-dictionary/>

<https://leetcode.com/problems/concatenated-words/>

DFS:

DFS আমি বেশী গুরুত্ব দিয়ে করেছি। Dynamic Programming এর Tabulation বা Bottom Up আমি ভালো পারি না, তাই DFS + Memoization বা Top Down নিয়ে বেশী কাজ করেছি।

<https://leetcode.com/problems/android-unlock-patterns/>

<https://leetcode.com/problems/cracking-the-safe/>

<https://leetcode.com/problems/swim-in-rising-water/>

<https://leetcode.com/problems/robot-room-cleaner/>

<https://leetcode.com/problems/distribute-coins-in-binary-tree/>

BFS:

BFS দিয়ে করলে সাধারণত Time Complexity বের করা DFS থেকে সহজ। তাই, যেই প্রবলেম DFS ও BFS দুইটা দিয়েই করা যায়, তখন আমি BFS দিয়ে করতাম।

<https://leetcode.com/problems/escape-a-large-maze/>

<https://leetcode.com/problems/shortest-path-to-get-all-keys/>

<https://leetcode.com/problems/trapping-rain-water-ii/>

<https://leetcode.com/problems/01-matrix/>

<https://leetcode.com/problems/open-the-lock/>

৪র্থ মাস: System Design, Dynamic Programming, Leadership Principle and More Practice

Leadership Principle:

Amazon এ Leadership Principle এ অনেক বেশী গুরুত্ব দেয়। আমার প্রথম টার্গেট ছিল

Amazon, তাই Leadership Principle এ অনেক সময় দিয়েছি। এই লিংকে দুইটায়

<https://medium.com/@scarletinked/are-you-the-leader-were-looking-for-interviewing-at-amazon-8301d787815d> ও

<https://kraftshala.com/what-questions-to-expect-in-amazon-interview/> মোটামুটি বিস্তারিত আছে।

System Design:

System Design এর জন্যও প্রস্তুতি নেয়া শুরু করি এই মাসে। প্রথমে

<https://www.educative.io/collection/5668639101419520/5649050225344512>

এবং <https://www.youtube.com/channel/UCRPMAqdtSgdoIpeef7iFsKw> এই দুইটা

পড়া শেষ করি। System Design এ যেই বিষয়গুলো মাথায় জানতে হবে, তা হল: Scalability:

Horizontal vs Vertical, Load Balancing, Caching, Data Partitioning, Indexing,

Proxy Server, Redundancy, Replication, Deduplication, SQL vs NoSQL, CAP

Theorem, Consistent Hashing, Long-Polling vs WebSockets. Grokking The System Design এ যে ডিজাইন আছে সেগুলো নিজে ডিজাইন করে কোন কোন components মিস করেছি সেগুলো দেখে নিতাম। একটা প্যাটার্নে নিজে বানিয়ে ওইটা সব সময় অনুসরণ করার চেষ্টা করতাম। প্রথমে, Storage, Bandwidth আর Traffic estimation। এরপরে Database ডিজাইন, High Level ডিজাইন। Client Side, Server, Storage, Load Balancer, Cache, Backup Storage এইগুলো একটার সাথে অন্যটার Data Flow ডিজাইন। এরপরে Data Storage কিভাবে করব, MySQL, Cassandra, Hadoop, MongoDB কোনটা কেন ভাল হবে, খারাপ হবে, Interviewer কে বার বার প্রশ্ন করে নিশ্চিত হতে হবে। এরপরে আসবে Data Sharding Technique, এখানেও Id based, User Based, Hash Based বিভিন্ন Technique এর Pros and Cons বলতে হবে। কোথায় কোথায় Load Balancer, Cache লাগবে বলতে হবে। অনেক জায়গায় connection এর জন্য http, long pooling, tcp, socket এইগুলো কোনটা কখন ভাল হবে বলতে হবে। এছাড়া আরো কিছু টার্ম যেমন, Consistent Hashing, CAP Theorem, Deduplication এইগুলো কখনও Applicable হলে Interviewer কে বলতে হবে।

Dynamic Programming:

আমি Tabulation বা Bottom-Up DP ভালো পারি না, তাই Top-Down, Recursion + Memoization এ বেশী জোড় দিয়েছি। আমার টার্গেট থাকত, প্রথমে শুধু Recursion দিয়ে প্রবলেম সমাধান করা, পরে Memoization যুক্ত করে Optimization করা। আবার প্রথম থেকেই Memoization চিন্তা করলে অনেক সময় আমার ঝামেলা লেগে যায়। নিচে কিছু প্রবলেম আছে, যেগুলো Bottom-Up Approach এ করা যায়, Recursion দিয়েও করা যায়, Recursion + Memoization দিয়েও হয়।

Bottom-Up DP:

<https://leetcode.com/problems/regular-expression-matching/>

<https://leetcode.com/problems/distinct-subsequences/>

<https://leetcode.com/problems/burst-balloons/>

<https://leetcode.com/problems/decode-ways/>

<https://leetcode.com/problems/longest-palindromic-subsequence/>

<https://leetcode.com/problems/partition-to-k-equal-sum-subsets/>

Top-Down DP 1D:

<https://leetcode.com/problems/maximum-subarray/>

<https://leetcode.com/problems/maximum-product-subarray/>

<https://leetcode.com/problems/house-robber-ii/>

<https://leetcode.com/problems/minimum-cost-for-tickets/>

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

Top-Down DP 2D:

<https://leetcode.com/problems/maximal-rectangle/>

<https://leetcode.com/problems/palindromic-substrings/>

<https://leetcode.com/problems/longest-palindromic-substring/>

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown/>

ইন্টারভিউতে সাধারণত প্রতি রাউন্ডে ২ টা প্রশ্ন করে, দ্বিতীয় প্রশ্ন বেশীর ভাগ সময়ে প্রথম প্রশ্নের Optimization করতে বলা হয়। LeetCode এ অনেক প্রবলেমেরই ২ বা তার বেশী ভার্সন আছে। এই ধরনের প্রবলেম বেশী করে সমাধান করা ভাল, এতে ইন্টারভিউতে Follow Up প্রশ্নের আইডিয়া হয়।

Follow up question:

<https://leetcode.com/problems/word-break/> and

<https://leetcode.com/problems/word-break-ii/>

<https://leetcode.com/problems/word-search-i/> and

<https://leetcode.com/problems/word-search-ii/>

<https://leetcode.com/problems/unique-paths/> and

<https://leetcode.com/problems/unique-paths-ii/>

<https://leetcode.com/problems/range-sum-query-mutable/> and

<https://leetcode.com/problems/range-sum-query-2d-mutable/>

<https://leetcode.com/problems/course-schedule/> and

<https://leetcode.com/problems/course-schedule-ii/> (চলবে)

পর্ব-৩

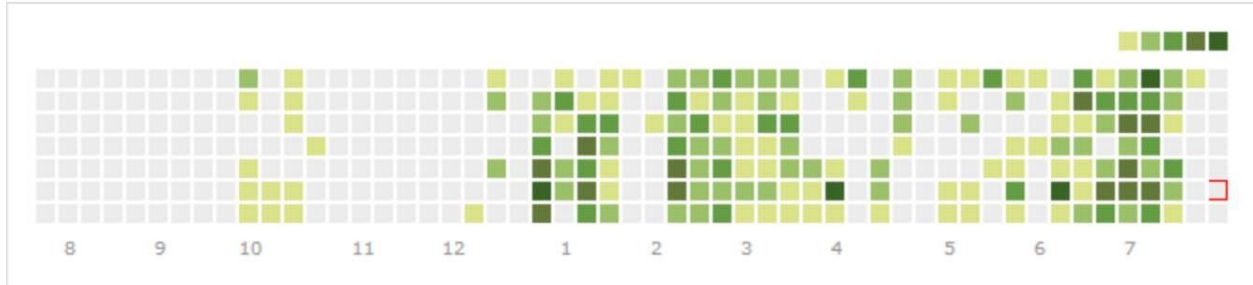
<https://www.facebook.com/notes/sabir-ismail/amazon-%E0%A6%93-google-%E0%A6%9A%E0%A6%BE%E0%A6%95%E0%A6%B0%E0%A6%BF%E0%A6%Bo-%E0%A6%AA%E0%A7%8D%E0%A6%B0%E0%A6%B8%E0%A7%8D%E0%A6%A4%E0%A7%81%E0%A6%A4%E0%A6%BF-%E0%A6%AA%E0%A6%B0%E0%A7%8D%E0%A6%AC-%E0%A7%A9/10157715212014674/>

Amazon ও Google চাকরির সুযোগ পাওয়ার প্রস্তুতি পর্ব-৫

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 2 MINUTES

৫ম মাসঃ API Design, More Pramp, More Practice

এই মাসে কিছু অনিবার্য কারণবশত একটু Off track ছিলাম, যাই হোক পরে আবার On Track হয়েছি। প্রস্তুতিতে জোয়ার-ভাটা থাকাও স্বাভাবিক।



API Design:

API Design প্রশ্ন খুব কমন। অনেক প্রবলেমে API Call করে ডাটা নিতে হয়। API Design এ Interviewer কে অনেক প্রশ্ন করতে হয়।

<https://leetcode.com/problems/add-and-search-word-data-structure-design/>

<https://leetcode.com/problems/flatten-2d-vector/>

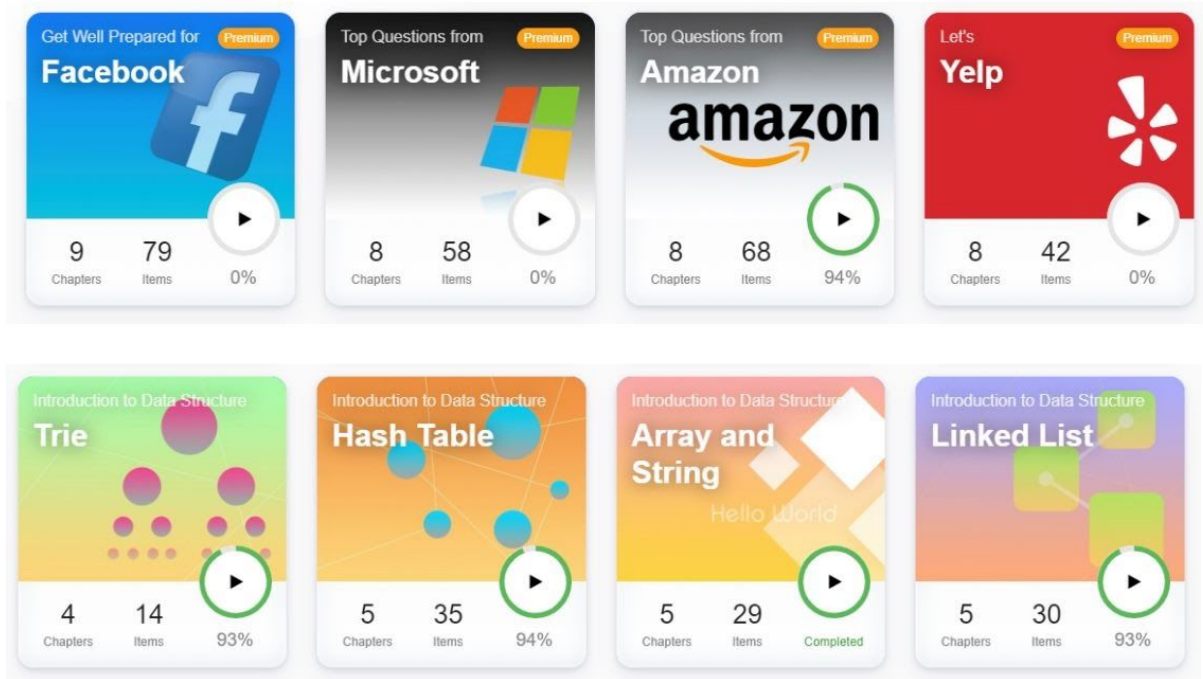
<https://leetcode.com/problems/serialize-and-deserialize-binary-tree/>

<https://leetcode.com/problems/logger-rate-limiter/>

<https://leetcode.com/problems/design-hit-counter/>

৬ষ্ঠ মাস এবং Check Local Lasting:



এই মাসে LeetCode Premium Account এর বিভিন্ন কোম্পানির জন্য আলাদা Preparation এবং বিষয় ভিত্তিক Preparation এর কোর্সগুলো করেছি





Pramp এ নিয়মিত ইন্টারভিউ দিতাম। LeetCode এ বিভিন্ন কোম্পানির Top Questions প্রবলেম থেকে Amazon ও Google এর জন্য প্রায় সবগুলো সমাধান করেছি, আগের করা থাকলেও আবার করেছি। এছাড়া Google ও Amazon tag দেয়া প্রবলেম থেকে Most Frequent গুলো করেছি।

Top Hits

 Top 100 Liked Questions

 Top Amazon Questions 

 Top Facebook Questions 

 Top Google Questions 

 Top Interview Questions

 Top LinkedIn Questions 

Companies

Google 750

Amazon 581

Facebook 459

Microsoft 403

Bloomberg 277

Uber 274

Apple 254

Adobe 230

LinkedIn 151

Yahoo 144

Alibaba 110

Snapchat 92

Oracle 90

Airbnb 86

eBay 85

Twitter 85

Goldman Sachs 83

অনেক প্রবলেম TAG করেও রাখতাম, ওইগুলো আবার Revise দিয়েছি। অনেক প্রবলেম আমি হয়ত সমাধান দেখে আগে করেছি, এখন আবার নিজে নিজে করার চেষ্টা করতাম।

LeetCode এ Discuss Section এ Interview Experience

<https://leetcode.com/discuss/interview-experience?currentPage=1&orderBy=hot&query=> থেকে Google আর Amazon এর সব Experience পড়েছি। এমন না যে, প্রবলেম কমন পড়বে, এইটা করায় মোটামুটি আইডিয়া পাওয়া যায়, ইন্টারভিউ কেমন হতে পারে। কোন প্রবলেম Interesting মনে হলে সমাধান করেছি। Glassdoor এও অনেকে Experience Share করে, <https://www.glassdoor.com/Interview/Google-Interview-Questions-E9079.htm> <https://www.glassdoor.com/Interview/Amazon-Interview-Questions-E6036.htm> . Glassdoor থেকেও গত ২ বছরে Google / Amazon এর প্রায় সব ইন্টারভিউ পড়েছি। (চলবে)

পর্ব-৪

<https://www.facebook.com/notes/sabir-ismail/amazon-%E0%A6%93-google-%>

Eo%A6%9A%Eo%A6%BE%Eo%A6%95%Eo%A6%Bo%Eo%A6%BF%Eo%A6%B
o-%Eo%A6%B8%Eo%A7%81%Eo%A6%AF%Eo%A7%8B%Eo%A6%97-%Eo%A
6%AA%Eo%A6%BE%Eo%A6%93%Eo%A7%9F%Eo%A6%BE%Eo%A6%Bo-%E
o%A6%AA%Eo%A7%8D%Eo%A6%Bo%Eo%A6%B8%Eo%A7%8D%Eo%A6%A4
%Eo%A7%81%Eo%A6%A4%Eo%A6%BF-%Eo%A6%AA%Eo%A6%Bo%Eo%A7
%8D%Eo%A6%AC-%Eo%A7%AA/10157715222659674/

Amazon ও Google চাকরির সুযোগ

পাওয়ার প্রস্তুতি পর্ব-৬

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 2 MINUTES

কোডিং:

Phone Screen এ Shared Doc আর Onsite এ Whiteboard এ Code লিখতে হয়। তাই
কাগজ-কলমে Code লিখে নিজে নিজে লাইন বাই লাইন Code Execute করার প্র্যাকটিস করতে হবে।
আমি প্রতিদিন ৪/৫ টা Code কাগজ-কলমে Execute করতাম, যখন মনে হত Code ঠিক আছে, তখন
LeetCode এ লিখে Run না করেই Submit দিতাম। Accept না হলে আবার কাগজ-কলমে ভুল ঠিক
করার চেষ্টা করতাম। আমি যেহেতু Whiteboard এ ক্লাস নিয়ে অভ্যস্ত তাই, Whiteboard বেশী
প্র্যাকটিস দিতে হয় নাই, ইন্টারভিউর ৭/৮ দিন আগে Whiteboard কিনে প্র্যাকটিস করেছি।
Whiteboard এ প্র্যাকটিসও করাও জরুরী। অনেকে Whiteboard এ লেখা বেশী ছোট বা বড় করে
ফেলে, আবার Whiteboard এর কোন জায়গা থেকে লেখা শুরু করা উচিত, এগুলো প্র্যাকটিস না দিলে
হবে না।

আগেই লিখেছি, Code Reusable, Modular, Production Ready হতে হয়। Null Value, Empty String, Negative/Zero Value এগুলো Exception Handling দিয়ে করা ভালো। Code করা শেষ হলে নিজে কিছু Test Case Generate করে Unit Test করা উচিত।

অনেক সময় বড় Code হলে Interviewer কে বল কিছু অংশ function লিখে Skip করা যেতে পারে। যেমন কোন প্রবলেমে হয়ত Tries লাগবে, সে ক্ষেত্রে Tries এর Insert(), Search() শুধু function নামে লিখে, পরে implement করা যেতে পারে। তবে অবশ্যই Interviewer বলতে হবে। Variable এর নামও meaningful এবং যাতে বেশী বড় না হয়, খেয়াল রাখতে হবে। কোন প্রবলেম করার সময় প্রথম থেকেই modular করার কথা চিন্তা করা উচিত। এমন না যে, প্রথমে Rough Code করে পরে modular করব, এটা করার সময় হবে না।

আমি অনেক প্রবলেম ৮/১০ বার করেও করেছি, বিভিন্ন ভাবে করেছি। এইটা আমার কাছে একটা প্রবলেমে বিভিন্ন ভাবে দেখতে সাহায্য করেছে এবং এক এক বার হয়ত এক এক ভাবে করার চেষ্টা করতাম। অনেক প্রবলেম রিভিউ দেওয়ার সময় হয়ত মনে মনে সমাধানটা চিন্তা করতাম। যদি Confident মনে হত তখন Code দেখে নিশ্চিত হয়ে নিতাম। আর Confident না হলে Code লিখে সমাধান করার চেষ্টা করতাম।

System Design:

Grokking the System Design কোর্স করলে মোটামুটি আইডিয়া হবে System Design সম্পর্কে। এরপরে আমি যা করেছি তা হল, নিজে একটা প্যাটার্ন তৈরী করেছি, কিছু Check List। 1. Requirements Analysis, 2. Storage, 3. Bandwidth and Traffic Estimation, 4. High Level Design, 5. Component Design, 6. Database, 7. Data Transfer, 8. Data Partitioning, 9. Cache, 10. Load Balancer, 11. Fault Tolerance and Replication, 12. Deduplication. এরপরে সব ডিজাইনে এই Check List অনুসরণ করতাম। কিছু কিছু Design এ

অবশ্য এগুলো ছাড়াও কিছু Component থাকে। যেমন, Uber এর মত Ride Share App ডিজাইনে Quad Tree Data Structure লাগতে পারে। System Design এ Interviewer এর সাথে খুব Engaging হতে হবে। যে কোন Component ফাইনাল করার আগে Interviewer কে Alternate অপশন ব্যাখ্যা করে কেন কোন Particular টা ব্যবহার করা ভালো বলতে হবে। যেমন messenger ডিজাইনে message কি ধরনের NoSQL এ থাকবে, Column Based, Document Based, Key-Value Based, Graph Based বলার পরে বলতে হবে কেন Column Based এ ক্ষেত্রে ভালো। Storage, Bandwidth, Traffic Estimation এ Realistic Assumption করতে হবে। Data Transfer এ Push, Pull, Hybrid অনেক উপায় আছে। Data Sharding এরও অনেক Technique আছে, যেমনঃ Instagram ডিজাইনে Photo ID, User ID, Timestamp অনেক ভাবে Data Sharding করা যায়। সবগুলো Interviewer এর সাথে আলোচনা করে একটা Select করতে হবে। Cache, Load Balancer অনেক জায়গায় বসানো যায়। কেন, কিভাবে Cache পুরো ডিজাইনকে Faster করে এবং কোন ধরনের Caching Algorithm ব্যবহার করা ভালো, এগুলো বলতে হবে। <http://highscalability.com/> এই সাইটে System Design এর অনেক তথ্য আছে।

<http://highscalability.com/amazon-architecture>

<http://highscalability.com/google-architecture>

<http://highscalability.com/youtube-architecture>

[http://highscalability.com/blog/2016/6/27/how-facebook-live-streams-to-8000 oo-simultaneous-viewers.html](http://highscalability.com/blog/2016/6/27/how-facebook-live-streams-to-8000-oo-simultaneous-viewers.html)

<http://highscalability.com/scaling-twitter-making-twitter-10000-percent-faster>

<http://highscalability.com/blog/2014/2/26/the-whatsapp-architecture-facebook-bought-for-19-billion.html>

<http://highscalability.com/blog/2015/9/14/how-uber-scales-their-real-time-market-platform.html>

<http://highscalability.com/blog/2011/12/19/how-twitter-stores-250-million-tweets-a-day-using-mysql.html>

<https://instagram-engineering.com/what-powers-instagram-hundreds-of-instances-dozens-of-technologies-adf2e22da2ad>

<https://www.youtube.com/watch?v=PE4gwstWhmc> (চলবে)

পর্ব-৫

<https://www.facebook.com/notes/sabir-ismail/amazon-%E0%A6%93-google-%E0%A6%9A%E0%A6%BE%E0%A6%95%E0%A6%B0%E0%A6%BF%E0%A6%B0-%E0%A6%B8%E0%A7%81%E0%A6%AF%E0%A7%8B%E0%A6%97-%E0%A6%AA%E0%A6%BE%E0%A6%93%E0%A7%9F%E0%A6%BE%E0%A6%B0-%E0%A6%AA%E0%A7%8D%E0%A6%B0%E0%A6%B8%E0%A7%8D%E0%A6%A4%E0%A7%81%E0%A6%A4%E0%A6%BF-%E0%A6%AA%E0%A6%B0%E0%A7%8D%E0%A6%AC-%E0%A7%AB/10157715236564674/>

Amazon ও Google চাকরির সুযোগ

পাওয়ার প্রস্তুতি শেষ পর্ব

SABIR ISMAIL · SUNDAY, SEPTEMBER 8, 2019 · READING TIME: 2 MINUTES

Object Oriented Design:

Object Oriented Design প্রশ্ন অনেক সময় আসে। আমি এটার জন্য তেমন ভালো করে কোন প্রস্তুতি নেই নাই। Cracking The Coding Interview বইয়ে একটা অধ্যায় আছে, Object Oriented Design নিয়ে, শুধু ওইটা দেখেছি। Recruiter এর সাথে কথা বলে ইন্টারভিউতে কি ধরনের প্রশ্ন আসবে নিশ্চিত হয়ে নেয়া ভালো। আমার কোন Object Oriented Design প্রশ্ন আসবে না, আমি আগেই জেনে নিয়েছিলাম.

Leadership Principle (LP) বা Cultural Fit:

LP প্রশ্নে সবসময় নিজের অভিজ্ঞতা বলতে হবে। যদি অভিজ্ঞতা নাও থাকে, তারপরও এই ধরনের Situation এ কি করতাম, সেটা বলতে হবে। অনেক সময় একই ধরনের প্রশ্ন পর পর রাউন্ডে আসতে পারে। সেক্ষেত্রে Interviewer কে বলতে হবে, আগের উত্তর দেয়া যাবে, নাকি ভিন্ন উত্তর দিতে হবে, অথবা অন্য প্রশ্নও করতে পারে।

Job Offer:

ইন্টারভিউ দেয়ার পরে Phone Screen এর ক্ষেত্রে সাধারণত ১ দিন পরেই Recruiter রা ফলাফল জানায়। Onsite এর ফলাফল অনেক কিছুর উপর নির্ভর করে। আমি Amazon ও Google দুইটাতেই ৭-১০ দিনের ভিতরে ফলাফল জানতে পেরেছি। আর একটা Start up থেকে ফলাফল পেতে প্রায় ১ মাস লেগেছে। প্রথমে Recruiter রা জানায় যে Offer Yes / No, কিন্তু Official Offer পেতে হয়ত আরো

সময় লাগে। এর মাঝে Team Matching এর ব্যাপার থাকে। ২/৩ জন Team Manager এর সাথে কথা বলে যে কোন Team Choose করা যায়। Team Choose হওয়ার পরে সাধারণত Official Offer দিয়ে একটা সময় দেয়া হয়, Offer Accept / Decline করার জন্য। অনেকের Multiple Offer থাকলে তখন চিন্তা ভাবনা করে সিদ্ধান্ত নিতে হয় বা অনেক সময় Compensation and Benefits নিয়ে দরদামও করা যায়।

Salary Negotiation:

Multiple Compitative Offer থাকলে Compensation and Benefits নিয়ে Negotiation করা যায়। কোন নির্দিষ্ট Job Position এর জন্য হয়ত একটা গড় Compensation থাকে, কিন্তু তার মানে এই না যে এটাই Fixed. এই সাইটে <https://www.levels.fyi/> এ বিষয়ের অনেক তথ্য আছে। প্রথম কথা হচ্ছে আগে, নিশ্চিত হতে হবে, আমাকে যে Compensation and Benefits দিচ্ছে সেটা Market Price এর সাথে সামঞ্জস্যপূর্ণ কি না। যদি না হয়, তাহলে অবশ্যই Recruiter কে জানাতে হবে। যদি মনে হয় Market Price এর সাথে সামঞ্জস্যপূর্ণ এবং হাতে আর কোন Job Offer নাই, সেক্ষেত্রে হয়ত Compensation and Benefits নিয়ে Negotiation করার সুযোগ কম।

পরিশিষ্ট:

আমি বলব চাকরির জন্য ৯৫% পরিশ্রম আর বাকি ৫% কপাল, কিন্তু ইন্টারভিউর সময় ৫% কপালের Weight ৯৫% আর ৯৫% পরিশ্রমের Weight ৫%। কারন ইন্টারভিউর দিন অনেক কিছুই হতে পারে। খুব কঠিন প্রশ্ন কিন্তু কমন পরতে পারে, আবার খুব সহ্য প্রশ্নে Brain Freeze হয়ে যেতে পারে, অন্য প্রার্থীদের তুলনামূলক ভালো/খারাপ হতে পারে। যেহেতু এই ৫% কপালের Weight ৯৫% এ আল্লাহ ছাড়া কারো হাত নেই, তাই বাকী ৯৫% পরিশ্রমের Weight ৫% কে যত Maximize করা যায়। আর এজন্য পরিশ্রমের বিকল্প নেই।

আমি যে প্রবলেমগুলোর লিংক দিয়েছি, এগুলো কোন Short List বা Suggestions না, আমি শুধুমাত্র প্রবলেমের বৈচিত্র্যতা বৃদ্ধানোর জন্য যে প্রবলেম আমার কাছে Interesting মনে হয়েছে সেগুলো দিয়েছি।

এই ৬ মাসে আমি নিয়মিত <https://www.quora.com/> আর <https://medium.com/> পড়তাম। <https://medium.com/> এর premium account ও কয়েক মাস ব্যবহার করেছি এবং এটা অনেক কাজে এসেছে।

৬ মাস আমি Full Time Employee এর মত সময় দিয়ে প্রস্তুতি নিয়েছি, প্রতিদিন ৭-৮ ঘণ্টা সময় দিয়েছি।

লেখায় সম্বোধন সূচক শব্দ ব্যবহার এবং অনিচ্ছাকৃত ভুল, আশা করি সবাই ক্ষমা সুন্দর দৃষ্টিতে দেখবেন।

লেখার কোন অংশ যদি আমার চাকরীর সাথে Conflicting কারো মনে হয়, তাহলে জানাবেন, আমি মুছে ফেলব।

(শেষ)

Reference:

বিবিধ লিঙ্ক:

1. <https://hbr.org/2016/09/a-guide-to-cold-emailing>, How to write cool email

2. <https://www.job-hunt.org/recruiters/linkedin-recruiters2.shtml>, How to add recruiter in LinkedIn
3. <https://www.linkedin.com/pulse/how-connect-recruiters-linkedin-using-my-bar-strategy-anne-pryor/>, How to add recruiter in LinkedIn
4. <https://yangshun.github.io/tech-interview-handbook/introduction>, Full preparation guide
5. <http://www.codespaghetti.com/interview-success>, How to success in Technical Interview
6. <http://blog.gainlo.co/index.php/category/system-design-interview-questions/>, System Design
7. [https://leetcode.com/problems/combination-sum/discuss/16502/A-general-approach-to-backtracking-questions-in-Java-\(Subsets-Permutations-Combination-Sum-Palindrome-Partitioning\)](https://leetcode.com/problems/combination-sum/discuss/16502/A-general-approach-to-backtracking-questions-in-Java-(Subsets-Permutations-Combination-Sum-Palindrome-Partitioning)), LeetCode, Backtracking Problem
8. <https://developers.google.com/machine-learning/glossary/>, Machine Learning Glossary

প্রস্তুতি ব্লগ:

1. <https://medium.com/@rachit138/how-i-cleared-the-amazon-sde-2-interview-f82a33706ff4>
2. <https://medium.com/hackernoon/how-i-got-hired-by-google-why-i-left-and-why-i-chose-to-join-okcoin-7af99c1d9c4f>

3. <https://medium.com/quick-code/top-tutorials-learn-data-structure-and-algorithm-for-an-interview-preparation-96e1f7518e26>
4. <https://medium.com/@jkarnows/google-software-engineer-interview-preparation-be8fa030ed0e>
5. <https://itnext.io/technical-interview-preparation-checklist-b000125f1535>
6. <https://blog.usejournal.com/how-i-got-into-google-161c97913b8b>
7. https://medium.com/@jayshah_84248/google-lost-a-chance-to-hire-me-finally-amazon-hired-me-e35076c73fe2
8. <https://medium.com/free-code-camp/top-tips-for-technical-interviews-from-9-of-instagrams-tech-girls-9d1b576bc287>
9. https://medium.com/@helen_zhang/the-4-week-plan-to-nailing-your-next-coding-technical-interview-internship-level-c5368c47e1d
10. <https://medium.com/swlh/ace-your-technical-interview-what-got-me-into-google-microsoft-and-more-119b21a9959f>
11. <https://medium.com/the-data-logs/my-first-technical-interview-549393efc407>
12. <https://medium.com/@jinzhe/my-google-interview-experience-51f716c7e578>
13. <https://medium.com/@vaibhavgupta/my-experience-of-interviewing-with-codenation-8dc06e832da4>
14. <https://blog.usejournal.com/from-hello-world-to-getting-a-job-at-google-23391dab849e>

পৰ্ব-৬<https://www.facebook.com/notes/sabir-ismail/amazon-%E0%A6%93-google-%E0%A6%9A%E0%A6%BE%E0%A6%95%E0%A6%Bo%E0%A6%BF%E0%A6%Bo-%E0%A6%B8%E0%A7%81%E0%A6%AF%E0%A7%8B%E0%A6%97-%E0%A6%AA%E0%A6%BE%E0%A6%93%E0%A7%9F%E0%A6%BE%E0%A6%Bo-%E0%A6%AA%E0%A7%8D%E0%A6%Bo%E0%A6%B8%E0%A7%8D%E0%A6%A4%E0%A7%81%E0%A6%A4%E0%A6%BF-%E0%A6%AA%E0%A6%Bo%E0%A7%8D%E0%A6%AC-%E0%A7%AC/10157715267349674/>

Github Link: https://github.com/sabir4063/my_preparation