

16. How to Prepare for Coding Interviews

Use LEETCODE.

WHY LEETCODE IS BETTER than other (free) platforms (in 2021):

- Problems were almost all asked in real interviews. No competitive programming or theoretical problems.
- Most of the problems have full explained solutions.
- For each problem there is a well-organized sub-section with a very active community.
- Hackerrank, codility, ... have too long problem statements or doesn't have clear solutions.
- Codeforces, Codechef, ... are for competitive programming.
- Most of the problems (probably all) on books like CTCI and EPI are on Leetcode. (Use those books just to learn the theory)

Leetcode alternatives:

- [Hackerrank](#) (used also for online assessment, contains also other programming topics)
- [Codility](#) (used also for online assessment)
- [Codesignal](#) (used also for online assessment)
- [HireVue](#) (used for online assessment)
- [binarysearch.com](#) (has a timer)
- [CodeForces](#) (Competitive programming)
- [Kattis](#) (Competitive programming)

How to prepare:

- Read the first 98 pages of CTCI that explains how interviews works.
- Watch a couple of videos on YouTube "*Google coding interview*" to get familiar with the context of coding interviews. (Do not watch too old videos, the interviews are changing over time)
- For each section of CTCI read the 2-3 theory pages and then do exercises on the respective <https://leetcode.com/explore/learn/> section. The optimal topic order for study should be this: Array, Hash table, Linked List, Stack & Queue, N-Trees, Binary Tree, Binary search trees, Trie, Heaps, Sorting, Searching, Recursion 1, Divide and Conquer (Recursion 2), Backtracking (Recursion 2), Dynamic Programming, Greedy, Graphs, Bit Manipulation, Math problems (combinations, permutation, prime numbers, probability, ...)
- Then do problems on the problem sets (Easy, Medium, Hard, Google, ...).
- Learn more theory on the books: Goodrich (in Python, Java or C++), Competitive Programming Handbook, Cormen.

General advices:

- Do problems on topics that you struggle at. Is not useful to do problems on topics you are already good.
- Be constant: Solve at least ONE problem a day. (Do something like Leetcode Daily challenge)
- Do one exercise a day on fundamental problems/algorithms (DFS, bin search, topological sort, ...).
- If a problem has too many dislikes read the statements and comments before solving it and waste your time.

- Once the problem is done, look at the discussion section and understand how other people solved it.
- Easy problems often are too easy. On the long term try to solve mostly medium and hard.
- Do problems on the interview lists of the company you want to enter more than once.
- Participate in Leetcode contests.
- Use a timer. (Chrome/Firefox Plugin for Leetcode: [Leetplug](#))
- Do a mock interview AT LEAST once a week and recreate as much as possible the interview environment (clothing, light, papers, materials, whiteboard, ...) to become comfortable.
- Use just 1 or 2 websites to prepare, otherwise you will end up doing all easy problems in a website, change to another website, do all the easy problems, then switch again and so on. You will end up doing just easy problems. Do a couple of problems on other websites just to become comfortable with the platform that is going to be used for online assessment (e.g. hackerrank, codility) and interviews (e.g. hirevue, coderpad).
- Write down the most interesting problems and techniques that you encounter. In general, medium problems are a composition of easy problems/techniques and hard problems are composition of medium problems/techniques.
- Do yourself a favor and take notes of the topics you are studying. During this time, you will be covering so many different subjects and tricks, and being a human being guarantees that you will forget the majority of them, so take notes and review them once in a while. Your notes are also a valuable resource for the next time that you are preparing for the interviews.

If you cannot solve a problem:

- Spend no more than 20-30 minutes without making progress. Just go look up the answer. Contrary to popular belief, most struggling past 30 minutes is pointless.
- Read just a little part of the solution and try to go on yourself. If you struggle, read a little bit more and so on, until you really cannot solve yourself.
- Always implement it yourself, don't just read the solution and copy and paste the code.
- Always mark it as something you need to try again. Wait at least a day and try to solve it fresh. If you fail, repeat at infinitum.

Aim to solve:

- 60 Easy problems (<10min)
- 100 Medium problems (<20min)
- 40 Hard problems (<30min)
- Ratio 3:5:2 (easy:medium:hard)

Problem Sets (to do in order):

- Leetcode Explore -> Interview -> Easy Collection
- Leetcode Explore -> Interview -> Medium Collection
- Leetcode Explore -> Interview -> Hard Collection
- Leetcode Explore -> Interview -> Specific company Collection (Premium)
- Leetcode BLIND 75 questions
- <https://seanprashad.com/leetcode-patterns/>
- Leetcode Curated Algo 170 (premium)
- Leetcode Mock Interview online assessment, phone, ... (Premium)
- [Leetcode Discussion: Microsoft-Online-Assessment-Questions](#)
- [Leetcode Discussion: Google-Online-Assessment-Questions](#)

- [Leetcode Discussion: Amazon-Online-Assessment-Questions](#)
- [Dynamic Programing](#)

Mock Interviews:

- [pramp.com](#) (free, also behavioral, frontend, ...)
- interviewing.io (paid) <https://www.youtube.com/c/interviewingio/videos>

Leetcode Problems ordered by score: [LINK](#) (Greater score means more difficult)