

I have been actively using leetcode as a platform to prepare for interviews for over 4 months now, and I feel I have learnt a few things along the way, that might or might not be useful for the other person looking for some direction, and getting a clear idea of how to progress on this journey.

[EASY]

- If you are just starting off, starting from easy problems is the best way to get in touch with your coding skills again, and/or build up new ones.
- As mentioned by other users, it's better to have a breadth-first approach to easy problems. Pick 5-6 questions from each topic and try to master the basics of the topic. I did a depth first of all the easy ones but looking back I feel it's better to move to the next topic once you feel confident in one.
- The purpose of the easy questions is to ingrain the basic concepts of a particular data structure or algorithm and if you look closely you will find a lot of questions asking for solutions testing your ability/understanding of language syntax, data structure core basics, etc, basically the "dirty" work, we so often don't give due attention. Leverage the same and get your hand dirty.
- You will also find a healthy mix of the use of two-three data structures or algorithms in some problems. Look for such problems, for they are a great way to test your understanding of how more than one DS work together to build a beautiful solution.
- That being said, I would recommend moving from the "easy" phase as soon as possible, for its the mediums that will be asked mostly in interviews.
- Must-do Easy questions [[Download the list here](#)] :
 - 13. Roman to Integer
 - 20. Valid Parentheses
 - 21. Merge Two Sorted Lists
 - 141. Linked List Cycle
 - 155. Min Stack
 - 169. Majority Element
 - 202. Happy Number
 - 204. Count Primes
 - 242. Valid Anagram
 - 371. Sum of Two Integers

- 88. Merge Sorted Array
- 108. Convert Sorted Array to Binary Search Tree
- 189. Rotate Array
- 205. Isomorphic Strings
- 226. Invert Binary Tree
- 448. Find All Numbers Disappeared in an Array
- 572. Subtree of Another Tree
- 557. Reverse Words in a String III
- 589. N-ary Tree Preorder Traversal
- 605. Can Place Flowers
- 617. Merge Two Binary Trees
- 665. Non-decreasing Array
- 669. Trim a Binary Search Tree
- 674. Longest Continuous Increasing Subsequence
- 703. Kth Largest Element in a Stream
- 705. Design HashSet
- 852. Peak Index in a Mountain Array
- 1160. Find Words That Can Be Formed by Characters

[MEDIUM]

- Once you feel confident enough with the basics, you can move onto the medium ones. Medium problems are pivotal in interview preparation since any decent interview will have at least one instance of medium problems. Moreover some of the advanced data structures and algos like graph, backtracking, dynamic programming are not well covered in the easy section, so the medium problems are a good stop to learn them.
- Again, breadth-first approach works wonder in this case as well. Pick like 10 problems from each topic, try to solve them to the best of your capabilities and practise of easy problems, and if you find it difficult to solve at first, take help from the discussion forum.
- What's important to note is the purpose of you doing N number of problems, should solely be to learn new techniques, be able to apply the learnt technique in similar problems later, and learn the basics of advanced data structure that you might have missed in the easy section.

- It's normal to not be able to solve medium level problems at first. Don't feel bad about taking help from the discussion forum, but try to learn the techniques as much as you can.
- *Do not cram the solution.* There is no point hoping you will remember the solution in an actual interview, because in a pressure situation it's not advisable to rely just on your memory. Rather place your bets on your acquired skills.
- While you are solving questions of a particular topic, make sure to give the theory a good read before or during the process of solving questions. It's important to know the basics of implementations of various data structures and algorithms like graph traversals, backtracking, memoization, trees, binary search etc. There are various resources available online for the same.
- Once you feel fairly confident after say, 60-70 mediums, it's always a good idea to start giving contests. Contests not only simulate the actual interview scenarios by having a time constraint, they also test your ability to be able to come up with solutions on your own without taking help from any other resources.
- Keep looking at better solutions posted by other users, compare your solution with them in terms of time and space complexities, know when to make the tradeoff between the two and ask for help whenever you feel stuck. Be open to learning and the rest will fall in place.
- A comprehensive list of must-do medium problems [[Download the list here](#)]:
 - 5. Longest Palindromic Substring
 - 11. Container With Most Water
 - 17. Letter Combinations of a Phone Number
 - 22. Generate Parentheses
 - 33. Search in Rotated Sorted Array
 - 34. Find First and Last Position of Element in Sorted Array
 - 46. Permutations
 - 24. Swap Nodes in Pairs
 - 48. Rotate Image
 - 49. Group Anagrams
 - 56. Merge Intervals
 - 75. Sort Colors
 - 94. Binary Tree Inorder Traversal

- 102. Binary Tree Level Order Traversal
- 106. Construct Binary Tree from Inorder and Postorder Traversal
- 117. Populating Next Right Pointers in Each Node II
- 142. Linked List Cycle II
- 150. Evaluate Reverse Polish Notation
- 151. Reverse Words in a String
- 166. Fraction to Recurring Decimal
- 179. Largest Number
- 200. Number of Islands
- 207. Course Schedule
- 208. Implement Trie (Prefix Tree)
- 215. Kth Largest Element in an Array
- 236. Lowest Common Ancestor of a Binary Tree
- 300. Longest Increasing Subsequence
- 328. Odd Even Linked List
- 338. Counting Bits
- 347. Top K Frequent Elements
- 355. Design Twitter
- 417. Pacific Atlantic Water Flow
- 421. Maximum XOR of Two Numbers in an Array
- 424. Longest Repeating Character Replacement
- 454. 4Sum II
- 529. Minesweeper
- 525. Contiguous Array
- 658. Find K Closest Elements
- 763. Partition Labels
- 767. Reorganize String
- 1171. Remove Zero Sum Consecutive Nodes from Linked List

[Other points to note]

- Leetcode is a wonderful community to learn, discuss and find support in others going through the same process. Leverage it.

- Don't shy away from being a part of discussions, ask for help, or ask questions from others just because you feel it sounds stupid.
- Constantly read articles on Leetcode, not only related to solutions but interview experiences, problems faced during the process etc. You can benefit so much from learning about stories of others here and also derive a sense of belonging to the whole process.
- Make notes of all what you have learnt from a question, a trick or technique for these will be handy when you will need to revise.
- It's okay to feel lost, confused or difficult to solve questions at times. Some questions are harder than others, some are trickier and some have no solid pattern to them. Don't get intimidated by the process. With time and practise everything gets better. Have patience and don't lose hope.

In the end, I want to mention that I am still not 'perfect' in solving problems, but leetcode have taught me a lot and I have definitely come a long way in this process, and with a little guidance, determination and patience anyone can.

Any updates to the list are welcomed, so are any questions and do share your experiences of your leetcode journey, for everyone can benefit from each other's experiences.

Good Luck and Happy Leetcoding!