How to revise for Coding Interviews in 15/30/45 days

Make a copy of this, turn it into a pdf, or store it on your computer. Read more, research more, explore more, and eventually choose what’s best for you!! Here are some points which we think would benefit you, let us know how we did @ [30dayscoding@gmail.com](mailto:30dayscoding@gmail.com). Good luck, you’ll do great!

## **Summary**

* Articles
* Videos
* Practice questions
* Solution reading
* Skimming through problems
* Practice lots of easy problems
* Time and space complexity revision

## **Data structures**

### Arrays

* + **60**% probability
  + Probably will be combined with another data structure.
  + Questions: [Find a peak element in an array](https://www.youtube.com/watch?v=a7D77DdhlFc&list=PLamzFoFxwoNjw4EpaVZzP-8lqWA9hOmnD), [Interview Question: Median of Arrays](https://www.youtube.com/watch?v=HGgdcKbC5ro&list=PLNmW52ef0uwvmnS0UQU4Qf3NvsEREGWoK&ab_channel=ByteByByte)

### Linked lists

* + **10**% probability
  + Practice/revise the generic things
    - Create, edit, delete
    - Reverse, join, multiple linked lists, etc.
    - [Video](https://www.youtube.com/watch?v=WwfhLC16bis&ab_channel=CSDojo)

### Trees

* + **50**% probability
  + Freecode camp notes: [Here](https://www.freecodecamp.org/news/all-you-need-to-know-about-tree-data-structures-bceacb85490c/)
  + Video series: [Here](https://www.ideserve.co.in/learn/tree-interview-questions)

### Graphs

* + **70**% probability
  + BFS, DFS, connected graph problems, travel salesman, 2d grid + graph
  + Intro: [Here](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2010/video-lectures/lecture-6-graph-theory-and-coloring/)
  + Playlist: [Here](https://www.youtube.com/playlist?list=PLDV1Zeh2NRsDGO4--qE8yH72HFL1Km93P)
  + Techie delight: [Here](https://www.techiedelight.com/Category/Graphs/)

### Tries

* + **20**% probability
  + Reading: [Here](https://www.freecodecamp.org/news/all-you-need-to-know-about-tree-data-structures-bceacb85490c/),
  + Traversals: [Here](https://www.youtube.com/watch?v=BHB0B1jFKQc)
  + Questions: [Here](https://www.ideserve.co.in/learn/tree-interview-questions)
  + Notes: [Here](http://www.bowdoin.edu/~ltoma/teaching/cs210/spring09/Slides/210-Trees.pdf)

## **Algorithms**

### Tools

* + Visualizer: <https://visualgo.net/en>

### DFS, BFS

* + Video: [Here](https://www.youtube.com/watch?v=7fujbpJ0LB4)
  + Video: [Here](https://www.youtube.com/watch?v=oDqjPvD54Ss)
  + Techie delight reading: [Here](https://www.techiedelight.com/depth-first-search-dfs-vs-breadth-first-search-bfs/)
  + BFS, DFS patterns 1: [Here](https://medium.com/leetcode-patterns/leetcode-pattern-1-bfs-dfs-25-of-the-problems-part-1-519450a84353)
  + BFS, DFS patterns 2: [Here](http://medium.com/leetcode-patterns/leetcode-pattern-2-dfs-bfs-25-of-the-problems-part-2-a5b269597f52)

### Dynamic programming

* + Coding ninjas - path: [Here](https://dynalist.io/d/FAnK1zgptw2q1G-QW1E9xq5i)
  + Video: [Here](https://www.youtube.com/watch?v=GACU5omoOGY&list=PLpO3gASfJEIJRnNG4q6QoHAYAATo466a_) (hindi)
    - Solve recursion problems
    - Start with easy ones
    - Recursion + memoization
    - Exploring all cases + storing the repeated values
    - Think recursively rather than finding a pattern
    - Solve 20 simple problems again and again, rather than doing 100
  + Medium article: [Here](https://medium.com/swlh/understanding-recursion-memoization-and-dynamic-programming-3-sides-of-the-same-coin-8c1f57ee5604)

### Binary search

* + Finite space of things, sorted
  + Video: [Here](https://www.youtube.com/watch?v=j5uXyPJ0Pew&list=PL2_aWCzGMAwL3ldWlrii6YeLszojgH77j)

### Topological sorting

* + Graph where you have one vertex as the root, basically a family tree
  + Geeks for geeks: [Here](https://www.geeksforgeeks.org/topological-sorting/)
  + William Fiset: [Here](https://www.youtube.com/watch?v=eL-KzMXSXXI&ab_channel=WilliamFiset)

## 

## **More algorithms**

* Bellman Ford
* Dijkstra's
* Floyd Warshall
* Ford Fulkerson
* Kahn's
* kosaraju's
* Kruskal's
* Prim's
* Tarjan's

## **Coding Patterns**

* 14 patterns: [Here](https://medium.com/hackernoon/14-patterns-to-ace-any-coding-interview-question-c5bb3357f6ed)
* Binary search
* 2 pointer pattern
* DFS, BFS
* Finding min/max - dynamic programming
* Tree traversals - inorder, postorder, preorder
* Heaps, priority queue for storing max things
  + Graph traversal with conditions
  + Keeping the max of something in trees
  + Longest shortest path
* DFS, BFS in 2d matrix
  + Iterative way of going from top left to bottom right
  + Make a stack/queue, add the root. Pop it -> add neighbors and then explore one by one
  + Return when you find something
  + Example problems: Word search
* Travel salesman
* Connected components in a graph
* Combination, permutation, combination sum - Backtracking
  + Include, recurse, exclude pattern
  + Kevin naughton video: Here
  + Template for solving backtracking problems: [Here](https://leetcode.com/problems/subsets/discuss/187733/BackTrack-Template-Make-backtracking-Easy!!!)
* 2d matrix dynamic programming patterns - knapsack etc.
* Advanced binary search: [Here](https://www.youtube.com/watch?v=GU7DpgHINWQ)

## **Reading solutions**

* Read solutions when you’re not practicing or studying
* Helps you get a grasp of how similar questions look
* Understand before moving forward. Don’t just literally ‘read’
* Make notes while going through each one
  + What you learnt
  + Piece of code you liked
  + Something new you found
  + question/pattern type
* Techie delight - [Here](https://www.techiedelight.com/)

## 

## **Mock interviews**

* Interviewing.io: [Here](https://interviewing.io/)
* Leetcode premium mock interview
* Pramp: [Here](https://www.pramp.com/#/)