

Study Guide

This guide is intended as a starting point for you to prepare for a technical interview. If you feel like you don't know where to start, or feel overwhelmed by the amount of topics to cover, we suggest this checklist of resources (short-tutorials + exercises) to kick-off your preparation.

This coursework is designed for you to get a solid understanding of data structures and algorithms and assumes that you have knowledge in coding, but it does not assume that you have knowledge around LinkedLists, Binary Trees, Sorting algorithms or related.

- Each exercise could take you up to 1 hour. Total duration estimation is 30 hours.
- These solutions are written in Java, but you will be able to use your language of preference in an interview.

Let's get started, but first...Something to keep in mind at all times is how to analyze how “good” a solution is. One way to measure that is to look at how long it takes for your solution to complete. Watch this video to get familiar with Big O notation <https://www.youtube.com/watch?v=v4cd1O4zkGw>.

Remember to always ask yourself, 'How long does this code take to run?' after solving the exercises here.

- Topic 1 → Arrays & Strings
 - [Warm Up - Learn how to use Hackerrank] [A Very Big Sum](#)
 - [Designer PDF Viewer](#)
 - [Left Rotation](#)
 - [Sparse Arrays](#)
- Topic 2 → Lists
 - Pre work: You need to become familiar with what a LinkedList is and how they work
 - [Read this](#)
 - [View this video](#)
 - [Take a look at this tutorial](#)
 - Exercises:
 - [Warm up] [Insert a Node at the tail of a LinkedList](#)
 - [Insert a Node at a position given in a list](#)

- [Merge two sorted Linked List](#)
 - Cycle Detection <https://www.hackerrank.com/challenges/detect-whether-a-linked-list-contains-a-cycle/problem>
- Topic 3 → Stacks & Queues
 - Pre work
 - Watch this video <https://www.hackerrank.com/challenges/waiter/problem>
 - Take a quick look to this tutorial https://www.tutorialspoint.com/data_structures_algorithms/stack_algorithm.htm
 - Exercises:
 - Balanced Brackets <https://www.hackerrank.com/challenges/balanced-brackets/problem>
 - Queue using two stacks <https://www.hackerrank.com/challenges/queue-using-two-stacks/problem>
 - Waiter <https://www.hackerrank.com/challenges/waiter/problem>
- Topic 4 → Hash & Maps
 - Pre work
 - Watch this video <https://www.youtube.com/watch?v=shs0KM3wKv8>
 - Read this resource <https://www.interviewbit.com/courses/programming/topics/hashing/>
 - Review these concepts in time and space complexity <https://www.interviewbit.com/courses/programming/topics/time-complexity/>
 - Exercises
 - Migratory birds <https://www.hackerrank.com/challenges/migratory-birds/problem>
 - Ice Cream Parlor <https://www.hackerrank.com/challenges/icecream-parlor/problem>
 - Missing Numbers <https://www.hackerrank.com/challenges/missing-numbers/problem>
 - Colorful Number [Colorful Number](#) - This one is hard. If you get stuck, refer to our proposed solution.
- Topic 5 → Sorting Algorithms
 - Pre-work
 - Bubble Sort <https://www.youtube.com/watch?v=6Gv8vg0kcHc>
 - Merge Sort <https://www.youtube.com/watch?v=KF2j-9iSf4Q>
 - Quick Sort [Algorithms: Quicksort](#)

- Exercises
 - Insertion Sort part 1 <https://www.hackerrank.com/challenges/insertionsort1/problem>
 - Insertion Sort part 2 <https://www.hackerrank.com/challenges/insertionsort2>
 - Correctness and the Loop Invariant <https://www.hackerrank.com/challenges/correctness-invariant/problem>
 - Quicksort part 1 <https://www.hackerrank.com/challenges/quicksort1/problem>
 - Running Time of Algorithms <https://www.hackerrank.com/challenges/runningtime/problem>
 - Quicksort part 2 <https://www.hackerrank.com/challenges/quicksort2/problem>
- Topic 6 → Trees
 - Theory
 - Watch this video to become familiar with Trees <https://www.youtube.com/watch?v=oSWTXtMgIKE&t=138s>
 - Watch this video about Binary Search Tree https://www.youtube.com/watch?v=i_Q0v_Ct5IY
 - Watch this video about Heaps to do the last exercise recommended <https://www.youtube.com/watch?v=t0Cq6tVNRBA&t=269s>
 - Exercises
 - Preorder Traversal <https://www.hackerrank.com/challenges/tree-preorder-traversal/problem>
 - Postorder Traversal <https://www.hackerrank.com/challenges/tree-postorder-traversal/problem>
 - Binary Tree Insertion <https://www.hackerrank.com/challenges/binary-search-tree-insertion/problem>
 - Height of a Binary Tree <https://www.hackerrank.com/challenges/tree-height-of-a-binary-tree/problem>
 - QHeap1 <https://www.hackerrank.com/challenges/qheap1/problem>
 - SwapNodes <https://www.hackerrank.com/challenges/swap-nodes-algo/problem>
- Topic 7 → Graphs (BFS & DFS)
 - Theory
 - Watch this video to understand what is a graph and how to traverse it <https://www.youtube.com/watch?v=zaBhtODEL0w>
 - Exercises
 - Breath First Search <https://www.hackerrank.com/challenges/bfsshortreach/problem>

- Snakes and Ladders <https://www.hackerrank.com/challenges/the-quickest-way-up/problem>
- Even Tree <https://www.hackerrank.com/challenges/even-tree/problem>
- Topic 8 → Recursion
 - Theory
 - Watch this video to review concepts on recursion <https://www.youtube.com/watch?v=KEEK7Me-ms>
 - Exercises
 - Print 1 to N recursively <https://www.hackerrank.com/contests/recursion/challenges/print-1-to-n-recursively>
 - Fibonacci Numbers <https://www.hackerrank.com/challenges/ctci-fibonacci-numbers/problem>
 - Factorial <https://www.hackerrank.com/challenges/30-recursion/problem>

SOLUTIONS

All solutions are available in this public repository: <https://github.com/lolapriego/coursework>