To crack interviews, you need three things:

1. Good understanding of data structures and algorithms
2. Good understanding of system design
3. Good practice on sites like topcoder, hackerrank, leetcode, etc.

You can start with “**Problem Solving in Data Structures and Algorithms”** written in various languages in C, C++, Java, C#, Python.

* Written for **interview point of view**
* Last chapters have **system design**.
* **Day 0**

Stick to a programming language like C/C++ or Java.

Make sure that you are comfortable with pointers/objects.

**Day 1**

Understand the concept of **Algorithmic Complexity**

* Skip the theory for now, but for every piece of code you write, you should be able to derive both time and space complexity.

**Day 2 – 10** Let’s start with some simple data structures

1. Arrays
2. Linked Lists
3. Strings
4. Stacks
5. Queues

Understand their basic operations (insert, delete, search, traverse) and their complexity, and code them all..

**Day 11 – 25** Let’s start with some simple algorithms

1. Sorting: Insertion, Merge, Quick, Heap, Bucket, Counting, Radix, External
2. Search: Linear, Binary (along with its variants)
3. Prime Numbers: Sieve of Erastosthenes, Primality Test
4. Strings: String searching, LCS, Palinedrome detection
5. Miscellaneous: Euclidean algorithm, Matrix multiplication, Fibonacci numbers, Pascal’s Triangle, Max Subarray Problem

**Day 25 – 60** Once you are comfortable with everything above, start doing problems from

1. CTCI
2. Elements of Programming Interviews
3. Programming Interviews Exposed: Secrets to Landing Your Next Job
4. GeeksforGeeks
5. HackerRank
6. InterviewBit

Stick to chapters of arrays, linked lists, strings, stacks, queues, and complexity.

**Day 51 – 60** Let’s learn some non-linear data structures

1. Tree: Binary, Binary Search, Tree traversals, Lowest common ancestor, Depth, Height and Diameter, Finding k-th smallest element
2. Heaps
3. Hash Table: 4 sum problem, checking if soduku solution is valid
4. Graph: BFS, DFS, Topological Sorting, Minimum Spaning Tree, Shortest Path Problem

**Day 61 – 90:** Refer to the previous resources and start doing problems from trees, hash tables, heaps and graphs

**Day 91 – 100:** Understand Computation Complexity theory

NP-completeness

Knapsack problem

Travelling Salesman

SAT problem and so on.