convenience, I'll use DSA). There will be many questions for a beginner like how to start learning DSA, as there are many concepts involved and he/she might get confused at the start. In this post, we are going to see a roadmap for learning DSA, which worked out for my friends. I am not saying that this is the perfect roadmap for DSA. You can use your own plans also but this is just an idea. So let's get started.

LANGUAGE SELECTION

Pick any programming language of your choice. If you want to build DSA from scratch, I'll suggest C/C++.

LEARN THE BASICS

For a fantastic building, you need an extraordinary basement. Get a good understanding of pointers, structures, classes, dynamic memory allocation, and recursion. Learn how to find time and space complexities of algorithms and learn about the notations.

ARRAYS (DS)

Learn about what is an array, how do they work, how data is stored in an array, how memory is managed in an array, and all other things. After knowing the basics of arrays, learn about the operations that we can perform on arrays like, insertion, deletion, replacing, searching, etc.

Searching algorithms.

Learn basic searching algorithms such as, Linear search and Binary search, with their time complexities and use cases. There are many searching algorithms but learning these first will give a clear understanding of others.

Sorting algorithms.

Learn basic sorting algorithms that follow a NORMAL APPROACH such as selection sort, bubble sort, etc.

DIVIDE AND CONQUER (A)

Learn about Divide and Conquer methodology. Sorting algorithms such as merge sort, quick sort, and others follow this methodology.

This is an important programming paradigm.

LINKED LISTS (DS)

Learn about what is a linked list, how do they work, how data is stored, how memory is managed, and all other things. After knowing the basics of linked lists, learn about the managed, and all other things. After knowing the basics of linked lists, learn about the operations that we can perform on them like insertion, deletion, searching, traversing, reversing, etc. Learn about their applications. Also know the different types of linked lists.

STACKS AND QUEUES (DS)

Follow the same steps as mentioned for linked lists.

TREES AND GRAPHS (DS)

Now we are coming to the most important part. Learn about binary trees, binary search trees, traversal methods, the height of a tree, and all other operations. Learn about the Depth-first search and Breadth-first search.

The same goes for Graphs. A tree, in fact, is a special type of graph.

Almost all algorithmic concepts that come under trees will definitely come under graphs. Do an IN-DEPTH analysis for understanding these data structures.

GREEDY METHODOLOGY (A)

Greedy is an algorithmic paradigm. Almost 70 percent of algorithms under trees and graphs follow this methodology.

Greedy Algorithms

1.Dijkstra’s shortest path algorithm.

2.Prim’s and Kruskal's Minimum Spanning Tree.

3.Dial’s Algorithm (Optimized Dijkstra for small range weights)

And other algorithms that follow the greedy approach. These are very important algorithms. After learning, apply this concept in other kinds of data structures to get the solution to the problem in that specified data structure.

DYNAMIC PROGRAMMING

This is also an algorithmic paradigm. Give more concentration to this one, as this is the toughest one of all other paradigms. Know the basics of how this paradigm works and move to the understanding of these algorithms.

Algorithms

1.Traveling salesman problem

2.Coin change problem

3.Bit Masking.

Get a good grasp of these algorithms by practicing them again and again.

BACKTRACKING

Know the foundation of this algorithm and learn about the N-Queens Problem. Practice as many problems as you can

After learning, try to apply a problem solving technique in some other data structure. For example, try dynamic programming approach in solving problems in arrays.

Finally, I would like to conclude by saying that, knowing all this is not enough. You must do hard work and practice enough problems daily for mastering these techniques. There are