DSA Preparation Guide

Table of Contents

1. Arrays

2. Strings

3. Linked Lists

4. Stacks and Queues

5. Recursion and Backtracking

6. Trees

7. Graphs

8. Dynamic Programming

9. Greedy Algorithms

10. Sorting and Searching

11. Hashing

12. Bit Manipulation

13. Divide and Conquer

14. Math

15. Tries

16. Heaps

Arrays

Details:

- Traversing, Insertion, Deletion

- Searching (Linear, Binary Search)

- Sorting (Bubble, Merge, Quick)

- Applications (Kadane's Algorithm, Prefix Sums)

MCQs:

1. What is the time complexity of Binary Search?

a) O(n) b) O(log n) c) O(n^2) d) O(n log n)

Strings

Details:

- Pattern Matching (KMP, Rabin-Karp)

- Anagram Checks

- Longest Palindromic Substring

MCQs:

1. What is the time complexity of KMP?

a) O(m+n) b) O(m*n) c) O(m^2) d) O(n^2)

Linked Lists

Details:

- Singly and Doubly Linked Lists

- Cycle Detection (Floyd's Algorithm)

- Reversing a Linked List

MCQs:

1. Which of these is used to detect cycles?

a) Floyd's Algorithm b) Kruskal's Algorithm c) Dijkstra's Algorithm d) None

DSA Cheat Sheet

Arrays:

- Prefix Sum: Useful for range queries.

- Kadanes Algorithm: Max subarray sum in O(n).

Strings:

- KMP: Pattern Matching O(m+n).

- Rabin-Karp: Rolling Hash O(n+m).

Linked Lists:

- Reverse a list: Iterative or Recursive.

- Detect cycle: Floyds Cycle Algorithm.

Sorting:

- QuickSort: Avg O(n log n), worst O(n^2).

- MergeSort: Stable, O(n log n).

Dynamic Programming:

- Coin Change: dp[i] = min(dp[i], 1+dp[i-coin]).

- LCS: dp[i][j] = max(dp[i-1][j], dp[i][j-1]).

Graphs:

- BFS/DFS: O(V+E).

- Dijkstra: Shortest path O(V^2) or O((V+E) log V) with PQ.