**Additional Topics:**

A. Optimization problems

Note: In the greedy approach, we apply a predefined procedure over the entire input array.

Greedy Algorithms

**//** [**Striver**](https://takeuforward.org/interviews/strivers-sde-sheet-top-coding-interview-problems/)

1. N meetings in one room / Activity Selection
2. Minimum number of platforms required for a railway / Meeting Rooms II
3. Job sequencing Problem
4. Fractional Knapsack Problem
5. Greedy algorithm to find minimum number of coins

**//** [**Abdul Bari**](https://www.youtube.com/watch?v=ARvQcqJ_-NY)

1. Fractional Knapsack Problem
2. Job sequencing Problem
3. Optimal Merge Pattern / Minimum Cost to Connect Sticks
4. Huffman Coding
5. Minimum Spanning Tree: Prim’s / Kruskal’s Algorithm
6. Dijkstra: Single source shortest path algorithm

//Heaps / Sorting related mostly

//2 pointer / Union Find may show up too

Note: We find all possible solutions in dynamic programming and select the best solution. Time-consuming compared to Greedy. Uses a recursive algorithm and follows the principle of optimality (A problem can be solved by taking a sequence of decisions i.e. decision at each step. In greedy a single decision is defined and used throughout.)

Recursive/Top Down/Memoization (Values propagate back up, higher num to lower)

Tabulation/Bottom Up (lower num to higher)

Dynamic Programming

**//** [**Abdul Bari**](https://www.youtube.com/watch?v=ARvQcqJ_-NY)

1. Fibonacci
2. Multistage graph / 120. Triangle / 64. Minimum Path Sum
3. All Pairs Shortest Path / Floyd Warshall / 1462. Course Schedule IV

Topological Sort (Apply here:1462. Course Schedule IV)

Branch and Bound

Segment Trees