**O(c) < O (logn) < O() < O(n) < O(nlogn) < O() < O() < O() < O()** (Follow mam’s pdf)

1. **Quick Sort:-**

Time Complexity(Best): O(n log n)

Worst: O(n^2)  
Space Complexity: O(log n)

T(n)=2T(n/2)+n : Best  
T(n)=T(n-1)+n

1. **Merge Sort:-**

Time Complexity: O(n log n)

Space Complexity: O(n)

T(n)=2T(n/2)+n

1. **Insertion Sort:-**

Time Complexity: O()

Space Complexity: O(1)

1. **Heap Sort:-**  
   Time Complexity: O(n log n)

Space Complexity: O(log n)

1. **Selection Sort:-**  
   Time Complexity: O()

Space Complexity: O(1)

1. **Kruskal’s Algorithm:-**

Time Complexity: O(E log E) or O(V log V)

Space Complexity: O(V+E)

1. **Dijkstra’s Algorithm:-**  
   Time Complexity: O(E + V log V)

Space Complexity: O(V)

1. **Job Sequencing:-**  
   Time Complexity: O()

Space Complexity: O(n)

1. **Knapsack:-**  
   Time Complexity: O(n\*W)

Space Complexity: O(n\*W) + O(n)

1. **Radix Sort:-**  
   Time Complexity: O(n+k)

Space Complexity: O(n+k)

1. **Counting Sort:-**  
   Time Complexity: O(n+k)

Space Complexity: O(k)

1. **OBST:-**  
   Time Complexity: O()

Space Complexity: O()

1. **Floyd-Warshall:-**  
   Time Complexity: O()

Space Complexity: O()

1. **Longest Common Subsequence:-**  
   Time Complexity: O(m\*n)

Space Complexity: O(m\*n)

1. **Travelling Salesman Problem:-**  
   Time Complexity: O(m\*n)

Space Complexity: O(m\*n)

1. **N-Queens using Backtracking:-**  
   Time Complexity: O(n!)

Space Complexity: O()

1. **Hamiltonian Circuit Problems:-**  
   Time Complexity: O(n!)

Space Complexity: O(1)

1. **Sum of Subsets using Backtracking:-**  
   Time Complexity: O()

Space Complexity: O(n)

1. **15 Puzzle Problem:-**  
   Time Complexity: O()

Space Complexity: O(n)

1. **0/1 Knapsack using Branch & Bound, Fractional using Greedy:-**  
   Time Complexity: O()

Space Complexity: O(n)