ADS (2Marks) Mid-II

Unit-3

(1) Write the basic idea of merge sort.

The Merge sort algorithm is a divide-and-conquer algorithm that sorts an array by first breaking it down into smaller arrays, and then building the array back together the correct way so that it is sorted.

divide := TO divide the given unsorted list into two sub-lists.

conquer := TO sort two sublists recursively.

(2) What are the three main steps in the Divide and conquer method?

The three main steps are :=

divide := In this step we will divide the given unsorted list into two sublists with each n/2 elements

conquer := It sorts two sublists recursively

combine := We will merge the two sorted sublists into one sorted list.

(3) What is the time complexity of Merge sort, and why is it considered efficient for large data sets?

→ The time complexity of Merge sort is O(n log n) in all cases.

→ It is considered efficient for large datasets because it consistently divides data into smaller parts and merges them efficiently, ensuring stable and predictable performance even for large inputs.

(4) List the algorithms that uses Greedy Method.

Algorithm's that uses Greedy method are :=

(1) Kruskal's algorithm
(2) Prim's Algorithm
(3) Dijkstra's algorithm
(4) Job sequencing with deadlines
(5) Fractional knapsack problem.

(5) What is the difference between 0/1 knapsack problem and fractional knapsack problem.

* In 0/1 knapsack problems, items cannot be divided, each item is either taken completely or left.
* In Fractional knapsack, items can be divided, and fractions of items can be taken to maximize profit.

→ 0/1 knapsack uses dynamic programming, while fractional knapsack uses Greedy method

(6) Mention advantages of merge sort over quick sort

* Time complexity :- Merge sort has a worst-case time-complexity of $O(n \log n)$, while Quicksort can degrade to $O(n^2)$ in the worst case
* Linked lists := Merge sort works efficiently with linked lists, while Quick sort is better suited for arrays
* Stability :- Merge sort is a stable sorting algorithm, while Quick sort is not

(7) What is a Minimum cost spanning tree? Name two algorithms that are commonly used to find MSTs.

A Minimum Cost Spanning Tree (MCST) is a spanning tree of a connected, weighted graph that connects all the vertices with the minimum possible total edge cost without forming any cycles.

Algorithms used to find MCST:=

(1) Prim's algorithm
(2) Kruskal's algorithm.

Unit-4

(1) What is queen attack and what is ruled to follow for identifying diagonal attack in N-Queen problem?

In the N-Queen problem, a Queen attack occurs when two or more queens are placed in the same row, column or diagonal i.e., in such a way that they could attack.

→ The Rule to identify diagonal attack is -

Two queens placed at position (i, j) and (K, l) are on the same diagonal if

$$|i-K| = |j-l|.$$

(2) Differentiate Dijkstra's and Bellman Ford Algorithms.

(A)	Dijkstra's	Bellman Ford
(1)	Works only with positive weights.	Works with positive & negative weights
(2)	Uses Greedy method	Uses Dynamic programming
(3)	Cannot detect negative cycles	Can detect negative cycles

(3) Define Optimal Binary Search Tree?

An Optimal Binary Search Tree (OBST) is a binary Search tree built in such a way that the total Search cost (based on the frequency or Searching of each key) is minimum.

→ The way in which the cost of a binary search ~~tree~~ tree is known as an optimal binary search tree.

(4) Differentiate greedy and dynamic programming?

Greedy Method	Dynamic Programming
(1) It is a technique where at each step you choose the best possible option hoping it leads to optimal solution.	(1) It solves problems by dividing them into subproblems storing their results and ensuring optimal solution
(2) Ex:- Fractional knapsack	(2) Ex:- 0/1 knapsack

(5) Define Dominance rule?

The dominance rule in Dynamic Programming is used to simplify a problem by removing choices that does not lead to an optimal solution.

→ It helps to save time and effort by keeping only best options. In

→ In merging operation, if S^i contains 2 pairs $(P_j, W_j), (P_k, W_k)$ & $P_j \leq P_k$ & $W_j \geq W_k$ the discard the pair (P_j, W_j).

(6) List the applications of dynamic programming

* All pairs shortest path
* 0/1 knapsack problem
* Single source shortest path
* String editing
* Travelling Sales person
* Optimal binary Search tree

(7) What is the main constraint in the 0/1 knapsack problem?

→ In 0/1 knapsack, total weight of selected items ≤ knapsack capacity (W).

→ each item can be chosen only once (0 or 1)

Subject to constrains, $\sum_{i=1}^{n} w_i x_i \leq M$

$x_i = 1$ or 0.

(8) What is string editing? What are the operation used in string editing?

The problem of string editing is to identify minimum cost sequence of edit operations that will transform one string to another string.

Operations:= (1) Insertion := Adding character to string

(2) Deletion := Remove a character to string

(3) Replace := Replace one character with another.

(9) Explain how Backtracking is different from Dynamic programming.

* Backtracking tries all possible solutions and backtracks when there is no feasible solution in that way.
* Dynamic programming solves subproblems once and then uses their result to find an optimal solution efficiently.

(10) What is the chromatic number of a graph?

The chromatic number of a graph is the minimum number of colours needed to color the vertices of the graph so that no two adjacent vertices have the same colour.

Unit-5

(1) List out the applications of Branch & Bound

* Travelling Sales person
* 0/1 knapsack problem
* 15 puzzle problem
* job scheduling problem
* Assignment problem.

(2) What is the difference between the upper bound and lower bound in the content of Branch and Bound?

Upper Bound (UB)	Lower Bound (LB)
(1) The best known solution cost so far.	(1) The minimum possible cost that can be achieved from a partial solution
(2) Any partial solution with a cost higher than UB can be ignored (Pruned)	(2) If LB of a branch is greater than UB, that branch cannot given a better solution and is pruned

(3) Define an NP-complete problem and provide one example

A decision problem is NP-complete if it is in NP and every problem in NP is reducable to it in polynomial time. i.e., it can be verified in polynomial time. An NP-complete is exclusively a decision problem

Ex:- Travelling Sales Person- finding the shortest possible route to all cities exactly once & returns to the starting city

(4) Name the applications of backtracking?

* N-Queen problem
* Graph colouring
* Sum of subset problem.
* 0/1 knapsack problem
* Hamiltanian cycle.

(5) Write the formula for calculating upper bound and lower bound in the 0/1 knapsack branch and Bound approach.

→ Upper bound does not allows fractions
→ lower bound allows fractions.

$$\text{Bound} = P + (M - W_c) \times \frac{P_j}{w_j}$$

where, P = current profit
W_c = Current weight
M = knapsack capacity
P_j/w_j = profit / weight of next object.

(6) Differentiate between fractional and 0/1 knapsack problems.

* In Fractional knapsack, items can be divided, and fractions of items can be taken to maximize Profit
* In 0/1 knapsack problem, items cannot be divided; each item is either taken completely, or left
→ 0/1 knapsack uses dynamic programming, while fractional knapsack uses Greedy method

(7) What is meant by graph colouring?

Graph colouring :- Let G is a graph and 'm' is a given positive integer. We have to colour the graph in such a way that no two adjacent vertices have the same colour by using 'm' colours. This problem is called as 'm' colour problem (or) 'm' colourability decision problem

(8) What is the travelling sales person problem?

The travelling sales person (TSP) in Branch and Bound is a problem of finding a shortest possible route that visits each city exactly once and returns to the starting city.

→ Lower Bound (LB) is used to find the minimum possible cost from a partial route.

→ Each branch represents a possible tour (path)