NOTE:
1) **Section-A** is **compulsory**
2) Attempt any **four** questions from **Section-B** and any **two** questions from **Section-C**
3) Any missing data may be assumed appropriately

Section – A   [Marks: 02 each]

Q1.
- a) Define Hamiltonian cycle.
- b) Arrange following time complexities in decreasing order : 2^n, n^2, logn, n.
- c) State sum of subsets problem.
- d) Differentiate between greedy approach and dynamic programming.
- e) Give an example of in-place sorting algorithm.
- f) Discuss how traveling sales person problem can be solved using dynamic programming.
- g) Discuss one problem for which greedy approach does not give optimal solution.
- h) Give best case and worst case time complexity of merge sort.
- i) Discuss approximation algorithms.
- j) What is significance of polynomial-time reduction?

Section – B   [Marks: 05 each]

Q2. State 0/1 knapsack problem. Find optimal solution for knapsack having capacity=11, where values and weights of objects are {v1,v2,v3,v4,v5)={1,6,18,22,28} and (w1,w2,w3,w4,w5}={1,2,5,6,7} respectively.

Q3. Find minimum spanning using Prim's algorithm and Kruskal's algorithm for the following graph.

1

P.T.O
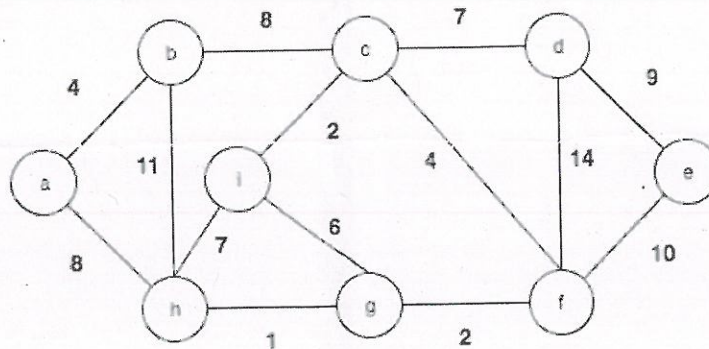
Q4. State 4-Queen's problem. Also discuss how backtracking approach helps in finding optimal solution for this problem.

Q5. Discuss different asymptotic notations to analyze complexity of an algorithm.

Q6. Differentiate between NP-hard and NP-complete problems. Give an example of each.

Section – C   [Marks: 10 each (05 for each sub-part, if any)]

Q7.(a) Compare Dijkstra's algorithm and Floyd–Warshall algorithm for finding shortest path.

(b) Find shortest path between each pair of vertices for following graph using Floyd-Warshall algorithm.
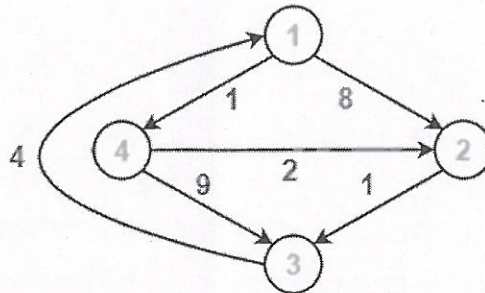
Q8. Write quick sort algorithm. Analyze its time complexity in best case and worst case inputs.

Q9. Discuss various pattern matching algorithms. Given the size of text as n characters and size of pattern as m characters, derive worst case time complexity of Robin-Karp algorithm and Brute Force algorithm.

****