DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PART - A

1. Define TSP problem and give any two applications.
2. Identify the purpose of Branch and Bound.
3. Name two problems that are classified as NP-complete.
4. Relate the CNF problem to NP-completeness.
5. Choose the correct class (P, NP, NP-hard, NP-complete) for the Hamiltonian Cycle problem.
6. Predict the time complexity behavior if NP-complete problems could be solved in polynomial time.
7. Tabulate the features of Ford-Fulkerson and Edmonds-Karp algorithms.
8. Relate graph algorithms to cybersecurity problems.
9. Compare the features of Ford-Fulkerson and Edmonds-Karp algorithms.
10. Infer the importance of the Floyd-Warshall algorithm.
11. Outline the principle of branch and bound.
12. Identify the representation of each state in the 15-puzzle problem
13. Define tractable problem with an example.
14. Compare NP-hard and NP-complete problems.
15. Identify any two polynomial time algorithms.
16. Define approximation algorithm.
17. Construct graph representation using adjacency matrix and adjacency list.
18. Differentiate between Bellman-Ford and Dijkstra’s algorithm.
19. State the time complexity of Floyd-Warshall algorithm.
20. Define a flow network.
21. List real-life application of the assignment problem.
22. Differentiate 0/1 Knapsack and Fractional Knapsack.
23. Infer the approximation ratio formula.
24. Define reduction in the context of NP-completeness.
25. Identify whether the Travelling Salesman Problem is NP-complete or NP-hard.
26. List real-world example of NP-complete problem
27. Outline bipartite graph with an example.
28. Infer the role of Graph Neural Networks (GNNs).
29. State the applications of dynamic graph algorithms.
30. Predict the role of graph algorithms in cybersecurity.

PART – B

1. Illustrate the working of a randomized approximation algorithm for the Bin Packing Problem with a suitable example.
2. Build a flowchart showing the steps of problem reduction from an NP-complete problem to another NP-complete problem.
3. Illustrate NP- algorithms, NP-hardness and NP-completeness with an example.
4. Demonstrate a polynomial-time approximation algorithm for the Travelling Salesman Problem (TSP).
5. Illustrate tractable and intractable problems using a Venn diagram with suitable examples.
6. Construct a dynamic graph algorithm to handle edge insertions and deletions efficiently.
7. Choose a graph algorithm applicable in cybersecurity to detect network vulnerabilities and explain its operation with an example.
8. Apply the Bellman-Ford algorithm to find the shortest path of a weighted directed graph with negative edges.
9. Solve a maximum bipartite matching problem using appropriate graph algorithm.
10. Use the Floyd-Warshall algorithm to compute all-pairs shortest paths for a given graph.
11. Demonstrate the Ford-Fulkerson method is applied in computing maximum network flow with an example.
12. Select the optimal assignment of 4 technicians to 4 machines in a factory such that the total repair time is minimized.

Technician	Machine 1	Machine 2	Machine 3	Machine 4
T1	5	8	6	7
T2	9	7	8	6
T3	4	6	7	5
T4	6	5	8	7

13. Select items for a hiking trip in a knapsack with capacity 18 kg to maximize total usefulness. Use this table:

Item	Weight (kg)	Value
T1	4	10
T2	8	40
T3	5	20
T4	3	30

14. Illustrate Assignment problem using branch and bound approach with an example.
15. Solve Travelling salesman problem using branch and bound approach with an example.
16. Solve the 15-puzzle problem using branch and bound technique.
17. Apply branch and bound to the Travelling Salesman Problem for 4 cities and the bound helps in pruning the search space.