**Unit I**

**Part – A Questions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Computer Science and Engineering** |  | |  |
|  | **23CS312-Design and analysis of Algorithm** |  | |  |
|  | **Third Semester** |  | |  |
| **Q.No** | **Questions** | **CO’s** | | **Bloom’s Level** |
| 1. | Define the term algorithm. | CO 1 | | K1 |
| 2. | Provide a visual representation illustrating the asymptotic notation of an algorithm, incorporating Big O, Omega, and Theta notations. | CO 1 | | K2 |
| 3. | Define time and space complexity. | CO 1 | | K1 |
| 4. | What are the types of algorithm efficiencies? | CO 1 | | K1 |
| 5. | What do you mean by best-case, worst-case and average case analysis of an algorithm? | CO 1 | | K1 |
| 6. | Define recurrence relations. | CO 1 | | K1 |
| 7. | What is the substitution method? | CO 1 | | K1 |
| 8. | What is lower bound in recurrence relation? | CO 1 | | K1 |
| 9. | What is the time complexity of linear search? | CO 1 | | K1 |
| 10. | What is the time complexity of binary search? | CO 1 | | K1 |
| 11. | What are the best-case, average-case, and worst-case time complexity of interpolation search? | CO 1 | | K1 |
| 12. | What is pattern matching? | CO 1 | | K1 |
| 13. | Define the basic idea behind the Naive string matching algorithm. | CO 1 | | K1 |
| 14. | Explain the concepts of time complexity and space complexity analysis in the context of insertion sort. | CO 1 | | K2 |
| 15. | Explain the concepts of time complexity and space complexity analysis in the context of heap sort. | CO 1 | | K2 |
| **Part – B** | | | | |
| 1. | Explain different types of algorithm efficiencies. (13 marks) | | CO 1 | K2 |
| 2. | Explain linear, binary and interpolation search with its algorithm. What is the time complexity required for each? (13 marks) | | CO 1 | K2 |
| 3. | Explain Naïve-string matching algorithm with an example. (13 marks) | | CO1 | K2 |
| 4. | Explain in detail about the types of hashing techniques.(13 marks) | | CO 1 | K2 |
| 5. | With a program explain insertion sort and analyze the space complexity of insertion sort. (13 marks) | | CO1 | K2 |
| 6. | Explain heap sort with an example. Write a heap sort algorithm and discuss it. (13 marks) | | CO1 | K2 |
| 7. | Write an algorithm for finding the maximum element of an array. Perform best, worst and average case complexity with appropriate order notations. (14 marks) | | CO1 | K2 |
| 8. | Explain the steps followed for Rabin-Karp-Algorithm for pattern searching with an example and discuss its time complexity. (14 marks) | | CO1 | K2 |

**UNIT II**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q.No** | **Questions** | **CO’s** | **Bloom’s Level** |
| 1. | Define graph and its types. | CO 2 | K1 |
| 2. | What is a complete graph? | CO 2 | K1 |
| 3. | What are directed graphs? | CO 2 | K1 |
| 4. | Define cycle. | CO 2 | K1 |
| 5. | Define strongly connected graph | CO 2 | K1 |
| 6. | Define indegree, outdegree in a graph. | CO 2 | K1 |
| 7. | What is the adjacent node? | CO 2 | K1 |
| 8. | What is a single source shortest path problem? - | CO 2 | K1 |
| 9. | What is breadth-first traversal? Give an example. | CO 2 | K1 |
| 10. | Illustrate the minimum spanning tree with an example. | CO 2 | K2 |
| 11. | Define Bellman-Ford algorithm. | CO 2 | K1 |
| 12. | How to calculate the efficiency of dijkstra’s algorithm. | CO 2 | K2 |
| 13. | What are shortest path algorithms? | CO 2 | K1 |
| 14. | What do you mean by perfect matching in a bipartite graph? | CO 2 | K1 |
| 15. | Define the constraint in the context of maximum flow problem. | CO 2 | K1 |
| **Part – B** | | | |
| 1. | Write an algorithm for all pairs shortest path algorithm and what are the time and space complexity of the algorithm (13 marks) | CO 2 | K2 |
| 2. | Apply Kruskal’s algorithm to find a minimum spanning tree of the following graph. (13 marks) | CO 2 | K3 |
| 3. | Explain Depth – first & Breadth – First Traversal algorithms. (13 marks) | CO 2 | K2 |
| 4. | Explain breadth first search algorithm and with its time and space complexity. (13 marks) | CO 2 | K2 |
| 5. | Write short notes on Biconnectivity. (13 marks) | CO 2 | K2 |
| 6. | Explain the algorithm for Maximum Bipartite Matching. (13 marks) | CO 2 | K2 |
| 7. | Explain the working of Bellman-Ford Algorithm to Detect the Negative cycle in the graph:  Lightbox | CO2 | K2 |
| 8. | Apply Dijkstra’s algorithm using the following graph. Find the shortest path between v1, v2, v3, v4, v5, v6 & v7. (14 marks) | CO2 | K3 |

**Unit – III**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q.No** | **Questions** | **CO’s** | **Bloom’s Level** |
| 1. | List the advantages of Divide and Conquer Algorithms. | CO 3 | K2 |
| 2. | Give the recurrence relation of divide-and-conquer? | CO 3 | K2 |
| 3. | Define Quick Sort. | CO 3 | K1 |
| 4. | List out the advantages of Quick Sort. | CO 3 | K2 |
| 5. | List the Steps in Merge Sort. | CO 3 | K2 |
| 6. | Describe the recurrence relation of merge sort? | CO 3 | K1 |
| 7. | List out the disadvantages of Quick Sort. | CO 3 | K2 |
| 8. | What is the time and space complexity of Merge sort? | CO 3 | K1 |
| 9. | What are the features of dynamic programming? | CO 3 | K1 |
| 10. | Write the difference between the Greedy method and Dynamic programming. | CO 3 | K1 |
| 11. | Is quick sort stable sorting algorithm? | CO 3 | K1 |
| 12. | State the general principle of greedy algorithms. | CO 3 | K2 |
| 13. | Define greedy methodology. | CO 3 | K1 |
| 14. | Define an optimization problem. | CO 3 | K1 |
| 15. | Differentiate PROS AND CONS of greedy algorithm. | CO 3 | K2 |
| **Part – B** | | | |
| 1. | Explain in detail in merge sort and give an example. (13 marks) | CO 3 | K2 |
| 2. | Sort the following set of elements using Quick sort 50,30,10,90,80,20,40,70. (13 marks) | CO 3 | K3 |
| 3. | Using the divide and conquer approach to find the maximum and minimum in a set of ‘n’ elements. Also find the recurrence relation for the number of elements compared and solve the same? (13 marks) | CO 3 | K3 |
| 4. | Explain in detail about Matrix Chain Multiplication. (13 marks) | CO 3 | K3 |
| 5. | (a). Differentiate between Divide and Conquer and Dynamic Programming. and (b). Differentiate Greedy method and Dynamic Programming . (13 marks) | CO 3 | K2 |
| 6. | Apply Dynamic programming, and solve a graph using the backward approach. (13 marks) | CO 3 | K3 |
| 7. | Briefly explain activity selection problem with suitable example. (14 marks) | CO3 | K3 |
| 8. | Apply the Huffman’s Algorithm for the given table. Construct the Huffman’s tree for the following data and obtain its Huffman’s Code. (14 marks)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | characters | A | B | C | D | E | F | | Probability | 0.5 | 0.35 | 0.5 | 0.1 | 0.4 | 0.2 | | CO 3 | K3 |

**Unit – IV**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q.No** | **Questions** | **CO’s** | **Bloom’s Level** |
| 1. | Differentiate backtracking and Exhaustive search. | CO 4 | K2 |
| 2. | What are the factors that influence the efficiency of the backtracking algorithm? | CO 4 | K1 |
| 3. | What is the n-queens problem? | CO 4 | K1 |
| 4. | Define the Hamiltonian cycle. | CO 4 | K1 |
| 5. | When can a node be terminated in the subset-sum problem? | CO 4 | K1 |
| 6. | How can the output of a backtracking algorithm be thought of? | CO 4 | K1 |
| 7. | Give a template for a generic backtracking algorithm. | CO 4 | K2 |
| 8. | State m color-ability decision problem. | CO 4 | K2 |
| 9. | Define implicit constraint. | CO 4 | K1 |
| 10. | What is a promising node in the state-space tree? | CO 4 | K1 |
| 11. | Define a state space tree. | CO 4 | K1 |
| 12. | Compare backtracking and branch and bound. | CO 4 | K2 |
| 13. | Give an example for a sum-of-subset problem. | CO 4 | K2 |
| 14. | Differentiate feasible solution and optimal solution. | CO 4 | K2 |
| 15. | Write the formula for the decision tree for searching a sorted array? | CO 4 | K1 |
| **Part – B** | | | |
| 1. | Write an algorithm to determine the Hamiltonian cycle in a given graph using backtracking. For the following graph determine the Hamiltonian cycle. (13 marks) | CO 4 | K3 |
| 2. | Explain the 4-queens problem using backtracking. Write the algorithms. Give the estimated cost for all possible solutions of the 4-queens problem. Specify the implicit and explicit constraints. (13 marks) | CO 4 | K3 |
| 3. | Explain the n-Queens problem and trace it for n=6. (13 marks) | CO 4 | K2 |
| 4. | How does backtracking work on the 8-Queens problem with suitable examples? (13 marks) | CO 4 | K2 |
| 5. | State the subset-sum problem and Complete state-space tree of the backtracking algorithm applied to the instance A= {3, 5, 6, 7} and d=15 of the subset-sum problem. (13 marks) | CO 4 | K3 |
| 6. | Explain in detail about knapsack problem with example.  (13 marks) | CO 4 | K2 |
| 7. | With an example, apply and explain Graph Colouring Algorithm. (14 marks) | CO 4 | K3 |
| 8. | Consider the travelling salesperson instances defined by the following cost matrix. (14 marks)    Apply the state space and show the reduced matrices corresponding to each of the node | CO 4 | K3 |

**Unit – V**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.No** | **Questions** | **CO’s** | | **Bloom’s Level** |
| 1. | Define Class NP. | CO 5 | | K1 |
| 2. | Define NP-complete with examples. | CO 5 | | K1 |
| 3. | List out the properties of NP-Hard Problems | CO 5 | | K1 |
| 4. | What is a venn diagram? | CO 5 | | K1 |
| 5. | List the types of venn diagrams. | CO 5 | | K1 |
| 6. | Define bounding. | CO 5 | | K1 |
| 7. | Define Problem Reduction. | CO 5 | | K2 |
| 8. | An NP-hard problem can be solved in deterministic polynomial time, how? | CO 5 | | K2 |
| 9. | Compare NP-Hard problems and NP-Complete. | CO 5 | | K2 |
| 10. | Define Bin-packing problem. | CO 5 | | K1 |
| 11. | What are tractable and intractable problems? | CO 5 | | K1 |
| 12. | Define Traveling salesman problem. | CO 5 | | K1 |
| 13. | What do you mean by primarily testing? | CO 6 | | K1 |
| 14. | What is Kth smallest number? | CO 6 | | K1 |
| 15. | How do quicksort algorithms use random pivoting? | CO 6 | | K2 |
| **Part – B** | | | | |
| 1. | What is class NP? Discuss about any five problems for which no polynomial time for TSP problem. (13 marks) | | CO 5 | K2 |
| 2. | Implement an algorithm for Knapsack problem using NP-Hard approach. (13 marks) | | CO 5 | K3 |
| 3. | Explain in detail about primality testing. (13 marks) | | CO 5 | K2 |
| 4. | Explain in detail about randomized quick sort with problem. (13 marks) | | CO 6 | K3 |
| 5. | Discuss in detail about Bin Packing problem. (13 marks) | | CO 6 | K2 |
| 6. | Discuss the approximation algorithm for NP-hard problems? (14 marks) | | CO 6 | K2 |
| 7. | Briefly explain about 3CNF problem with algorithm.(14 marks) | | CO 6 | K3 |