SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai - 600089

**FACULTY OF ENGINEERING AND TECHNOLOGY**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

****

**QUESTION BANK**

**DEGREE / BRANCH: B.Tech/CSE**

**IV SEMESTER**

**18CSC204J – Design and Analysis of Algorithms**

**Regulation – 2018**

**Academic Year 2021-2022**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

**Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai-600089**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**QUESTION BANK**

**SUBJECT : 18CSC204J – Design and Analysis of Algorithms**

**SEM/ YEAR: IV/ II**

**Course Outcomes**

**CO1:** Apply efficient algorithms to reduce space and time complexity of both recurrent and non-recurrent relations

**CO2:** Solve problems using divide and conquer approaches

**CO3:** Apply greedy and dynamic programming types techniques to solve polynomial time problems.

**CO4:** Create exponential problems using backtracking and branch and bound approaches.

**CO5:** Interpret various approximation algorithms and interpret solutions to evaluate P type, NP Type, NPC, NP Hard problems

**CO6:** Create algorithms that are efficient in space and time complexities by using divide conquer, greedy, backtracking technique

|  |  |  |  |
| --- | --- | --- | --- |
| **UNIT IV** | | | |
| **Introduction to backtracking - branch and bound, N queen’s problem – backtracking, Sum of subsets using backtracking, Complexity calculation of sum of subsets, Graph introduction, Hamiltonian circuit – backtracking, Branch and bound - Knapsack problem, Example and complexity calculation. Differentiate with dynamic and greedy, Travelling salesman problem using branch and bound, Travelling salesman problem using branch and bound example, Travelling salesman problem using branch and bound example, Time complexity calculation with an example, Graph algorithms, Depth first search and Breadth first search, Shortest path introduction, Floyd-Warshall Introduction, Floyd-Warshall with sample graph, Floyd-Warshall complexity** | | | |
| **PART-A (Multiple Choice Questions)** | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** |
| **1** | Which of the following is not a backtracking algorithm?  A) Knight tour problem  B) N queen problem  C) Tower of hanoi  D) M coloring problem  Answer: - C | CO 4 | BT 1 |
| **2** | Backtracking algorithm is implemented by constructing a tree of choices called as?  A) State-space tree  B) State-chart tree  C) Node tree  D) Backtracking tree  Answer: - A | CO 4 | BT 2 |
| **3** | What happens when the backtracking algorithm reaches a complete solution?  A) It backtracks to the root  B) It continues searching for other possible solutions  C) It traverses from a different route  D) Recursively traverses through the same route  Answer: - B | CO 4 | BT 1 |
| **4** | In what manner is a state-space tree for a backtracking algorithm constructed?  A) Depth-first search  B) Breadth-first search  C) Twice around the tree  D) Nearest neighbour first  Answer: - A | CO 4 | BT 2 |
| **5** | In general, backtracking can be used to solve?  A) Numerical problems  B) Exhaustive search  C) Combinatorial problems  D) Graph coloring problems  Answer: - C | CO 4 | BT 3 |
| **6** | Which one of the following is an application of the backtracking algorithm?  A) Finding the shortest path  B) Finding the efficient quantity to shop  C) Ludo  D) Crossword  Answer: - D | CO 4 | BT 1 |
| **7** | Who coined the term ‘backtracking’?  A) Lehmer  B) Donald  C) Ross  D) Ford  Answer: - A | CO 4 | BT 3 |
| **8** | The problem of finding a subset of positive integers whose sum is equal to a given positive integer is called as?  A) n- queen problem  B) subset sum problem  C) knapsack problem  D) hamiltonian circuit problem  Answer: - B | CO 4 | BT 2 |
| **9** | The problem of placing n queens in a chessboard such that no two queens attack each other is called as?  A) n-queen problem  B) eight queens puzzle  C) four queens puzzle  D) 1-queen problem  Answer: - A | CO 4 | BT 2 |
| **10** | In how many directions do queens attack each other?  A) 1  B) 2  C) 3  D) 4  Answer: - C | CO 4 | BT 4 |
| **11** | Placing n-queens so that no two queens attack each other is called?  A) n-queen’s problem  B) 8-queen’s problem  C) Hamiltonian circuit problem  D) subset sum problem  Answer: - A | CO 4 | BT 2 |
| **12** | Where is the n-queens problem implemented?  A) carom  B) chess  C) ludo  D) cards  Answer: - B | CO 4 | BT 1 |
| **13** | In the n-queen problem, how many values of n does not provide an optimal solution?  A) 1  B) 2  C) 3  D) 4  Answer: - B | CO 4 | BT 1 |
| **14** | Which of the following methods can be used to solve n-queen’s problem?  A) greedy algorithm  B) divide and conquer  C) iterative improvement  D) backtracking  Answer: - D | CO 4 | BT 1 |
| **15** | Which one of the following is a correct option that provides an optimal solution for 4-queens problem?  A) (3,1,4,2)  B) (2,3,1,4)  C) (4,3,2,1)  D) (4,2,3,1).  Answer: - A | CO 4 | BT 1 |
| **16** | How many possible solutions exist for an 8-queen problem?  A) 100  B) 98  C) 92  D) 88  Answer: - C | CO 4 | BT 2 |
| **17** | The Knapsack problem is an example of \_\_\_\_\_\_\_\_\_\_\_\_  A) Greedy algorithm  B) 2D dynamic programming  C) 1D dynamic programming  D) Divide and conquer  Answer: - B | CO 4 | BT 1 |
| **18** | Which of the following methods can be used to solve the Knapsack problem?  A) Brute force algorithm  B) Recursion  C) Dynamic programming  D) Brute force, Recursion and Dynamic Programming  Answer: - D | CO 4 | BT 1 |
| **19** | You are given a knapsack that can carry a maximum weight of 60. There are 4 items with weights {20, 30, 40, 70} and values {70, 80, 90, 200}. What is the maximum value of the items you can carry using the knapsack?  A) 160  B) 200  C) 170  D) 90  Answer: - A | CO 4 | BT 1 |
| **20** | What is the time complexity of the brute force algorithm used to solve the Knapsack problem?  A) O(n)  B) O(n!)  C) O(2n)  D) O(n3)  Answer: - C | CO 4 | BT 1 |
| **21** | Which of the following is/are property/properties of a dynamic programming problem?  A) Optimal substructure  B) Overlapping subproblems  C) Greedy approach  D) Both optimal substructure and overlapping subproblems  Answer: - D | CO 4 | BT 1 |
| **22** | If an optimal solution can be created for a problem by constructing optimal solutions for its subproblems, the problem possesses \_\_\_\_\_\_\_\_\_\_\_\_ property.  A) Overlapping subproblems  B) Optimal substructure  C) Memoization  D) Greedy  Answer: - B | CO 4 | BT 1 |
| **23** | If a problem can be broken into subproblems which are reused several times, the problem possesses \_\_\_\_\_\_\_\_\_\_\_\_ property.  A) Overlapping subproblems  B) Optimal substructure  C) Memoization  D) Greedy  Answer: - A | CO 4 | BT 2 |
| **24** | If a problem can be solved by combining optimal solutions to non-overlapping problems, the strategy is called \_\_\_\_\_\_\_\_\_\_\_\_\_  A) Dynamic programming  B) Greedy  C) Divide and conquer  D) Recursion  Answer: - C | CO 4 | BT 1 |
| **25** | When a top-down approach of dynamic programming is applied to a problem, it usually \_\_\_\_\_\_\_\_\_\_\_\_\_  A) Decreases both, the time complexity and the space complexity  B) Decreases the time complexity and increases the space complexity  C) Increases the time complexity and decreases the space complexity  D) Increases both, the time complexity and the space complexity  Answer: - B | CO 4 | BT 1 |
| **PART B (4 Marks)** | | | |
| **1** | What is meant by knapsack problem? | CO4 | BT 1 |
| **2** | Define fractional knapsack problem | CO4 | BT 1 |
| **3** | Write the running time of 0/1 knapsack problem. | CO4 | BT 3 |
| **4** | Write recurrence relation for 0/1 knapsack problem | CO4 | BT 3 |
| **5** | What is meant by travelling salesperson problem? | CO4 | BT 2 |
| **6** | What is the running time of dynamic programming TSP? | CO4 | BT 2 |
| **7** | State if backtracking always produces optimal solution | CO4 | BT 2 |
| **8** | Define backtracking. | CO4 | BT 1 |
| **9** | What are the two types of constraints used in backtracking? | CO4 | BT 1 |
| **10** | Define 8queens problem. 8. List out the application of backtracking | CO4 | BT 2 |
| **PART C (12 Marks)** | | | |
| **1** | Describe the travelling salesman problem and discuss how to solve it using dynamic programming? | CO4 | BT 2 |
| **2** | Apply backtracking technique to solve the following instance of the subset sum problem S = [1,3,4,5} and d=11 16 | CO4 | BT 3 |
| **3** | Explain subset-sum problem and discuss the possible solution strategies using backtracking | CO4 | BT 2 |
| **4** | Discuss the solution for knapsack problem using branch and bound technique. 12. What is branch and bound technique?Explain how knapsack problem could be solved using branch and bound technique.Solve the following instance of the knapsack problem by branch and bound algorithm for W=16 | CO4 | BT 3 |
| **5** | Apply backtracking to the problem of finding a Hamiltonian circuit for the following graph. | CO4 | BT 3 |

**Note:**

1. **BT Level –** Blooms Taxonomy Level
2. **CO – Course Outcomes**

BT1 – Remember BT2 – Understand BT3 – Apply BT4 – Analyze BT5 – Evaluate BT6 – Create