

**VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY NAMBUR-522508 ANDHRA PRADESH, INDIA**

**YEAR :** III B.Tech **SEMESTER:** II

**COURSE NAME:** DESIGN AND ANALYSIS OF ALGORITHMS

**COURSE CODE:** XXXXXXXX

**BRANCH:** CSM

**PREREQUISITE:** Basic Knowledge on Algorithms and Data Structures

**COURSE OBJECTIVE:**

* To familiarize students with various notations to represent Algorithms.
* To understand various asymptotic notations.
* To familiarize with various design methodologies.
* To learn solving problems by choosing appropriate design methodology.

**COURSE OUTCOMES:** Students will be able to:

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **OUTCOME** | **Cognitive Levels as per Bloom’s Taxonomy** | **Weightage (%)** |
| CO1 | Infer the divide-and-conquer paradigm and its context. Recite algorithms that employ this paradigm. Apply this paradigm to design algorithms for apt problems. Derive and solve recurrences describing the performance of divide-and-conquer algorithms. | L1, L2 | 20 |
| CO2 | Infer the greedy paradigm and its context. Recite algorithms that employ this paradigm. Apply this paradigm to design algorithms for apt problems. | L2, L3, L4 | 20 |
| CO3 | Infer the dynamic-programming paradigm and its context. Recite algorithms that employ this paradigm. Apply this paradigm to design algorithms for apt problems. | L1, L2, L3, L4 | 20 |
| CO4 | Infer the backtracking paradigm and its context. Recite algorithms that employ this paradigm. Apply this paradigm to design algorithms for apt problems. | L1, L2, L3, L4 | 20 |
| CO5 | Infer the branch and bound paradigm and its context. Recite algorithms that employ this paradigm. Apply this paradigm to design algorithms for apt problems | L1, L2, L3, L4 | 20 |

**WEIGHTAGE OF BLOOM’S LEGENDS & PERCENTAGE OF QUESTIONS IN EXAMINATIONS:**

L1 (Remembering) = 30 - 40%, L2 (Understanding) = 30 - 40%,

L3 (Applying) = 10 - 20 %, L4 (Analysing) = 10 - 20%,

Easy (%) = 15%-20%, Average (%) = 60% - 70%, Difficult (%) = 15% - 20%

TOTAL = L1 + L2 + L3 + L4 = 100% (on an average about 2 minutes per mark)

**Note:** This specification weightage in above shall be treated as a general guideline for students, teachers and paper setters. The actual distribution of marks in the question paper may vary slightly.

**DETAILED SYLLABUS:**

**UNIT - I 9 Hrs**

Introduction: Algorithm Definition, Algorithm Specification, Performance Analysis, Performance Measurement, Asymptotic notations. Divide and Conquer: General Method, Binary Search, Finding the Maximum and Minimum, Quick Sort.

**UNIT - II 10 Hrs**

The Greedy Method: The General Method, Knapsack Problem, Optimal Storage on Tapes Problem, Single Source Shortest Path Problem, Optimal Merge Patterns Problem.

**UNIT - III 12 Hrs**

Dynamic Programming: The General Method, 0/1 Knapsack Problem, Single Source Shortest Path – General Weights, All Pairs-Shortest Paths Problem, Traveling Salesperson Problem, String Editing Problem.

**UNIT – IV 10 Hrs**

Backtracking: The General Method, The N-Queens Problem, Sum of Subsets Problem, Graph Coloring Problem, Hamiltonian Cycles Problem.

**UNIT - V 10 Hrs**

Branch and Bound: The General Method, FIFO Branch-and-Bound, LC Branch-and-Bound, 0/1 Knapsack Problem, Traveling Salesperson Problem. NP-Hard and NP-Complete problems: Basic concepts, Cook’s Theorem.

**TEXT BOOKS:**

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, 2nd Edition, Universities Press.

**Reference Books:**

1. Harsh Bhasin, “Algorithms Design & Analysis”, Oxford University Press.

S. Sridhar, “Design and Analysis of Algorithms”, Oxford University Press.

**Web Resources:**

1.https://www.geeksforgeeks.org/fundamentals-of-algorithms/

2. https://www.javatpoint.com/daa-tutorial

3. https://nptel.ac.in/courses/106106131

4.https://www.tutorialspoint.com/design\_and\_analysis\_of\_algorithms/index.htm

**MICRO-SYLLABUS:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNIT I:** **9 Hrs**  **Introduction:** Algorithm Definition, Algorithm Specification, Performance Analysis, Performance Measurement, Asymptotic notation. Divide and Conquer: General Method, Binary Search, Finding the Maximum and Minimum, Quick Sort. | | | | | |
| **Unit** | | **Module** | **Micro content** | | **No of Hrs** |
| **I** | | Algorithm Analysis | Definition of Algorithm, Properties of algorithm | | 1 |
| Algorithm Specification – Pseudo code Conventions | | 1 |
| Performance Analysis – time and space complexity | | 1 |
| Performance Measurement – step count and frequency count | | 1 |
| Asymptotic Notations – Big Oh, Omega, Theta | | 1 |
| Divide and conquer | General Method | | 1 |
| Binary Search – Procedure, Example, Algorithm and Computing Time Complexity | | 1 |
| Finding the Maximum and Minimum - Procedure, Example, Algorithm and Computing Time Complexity | | 1 |
| Quick Sort - Procedure, Example, Algorithm and Computing Time Complexity | | 1 |
| **UNIT–II: 10 Hrs**  **The Greedy Method:** The General Method, Knapsack Problem, Job Sequencing with Deadlines Problem, Single Source Shortest Path Problem, Optimal Merge Patterns Problem. | | | | | |
| **Unit** | | **Module** | **Micro content** | | **No of Hrs** |
| **II** | | Greedy Method | General Method | | 1 |
| Knapsack Problem - Description, Example, Algorithm. | | 2 |
| Single Source Shortest Path Problem - Description, Example, Algorithm. | | 2 |
| Optimal Storage on Tapes Problem - Description, Example, Algorithm. | | 3 |
| Optimal Merge Patterns Problem - Description, Example, Algorithm. | | 2 |
|  | |
| **UNIT – III: 12 Hrs**  **Dynamic Programming:** The General Method, 0/1 Knapsack Problem, Traveling Salesperson Problem, All Pairs-Shortest Paths Problem, Traveling Salesperson Problem, String Editing Problem. | | | | | |
| **Unit** | **Module** | | **Micro content** | **No of Hrs** | |
| **III** | Dynamic Programming | | The General Method | 1 | |
| 0/1 Knapsack Problem - Description, Example. | 2 | |
| Single Source Shortest Path – General Weights - Description, Example. | 2 | |
| All Pairs-Shortest Paths Problem - Description, Example. | 2 | |
| Travelling Salesperson Problem - Description, Example. | 2 | |
| String Editing Problem - Description, Example. | 3 | |
| **UNIT – IV: 10 Hrs**  **Backtracking:** The General Method, The N-Queens Problem, Sum of Subsets Problem, Graph Coloring Problem, Hamiltonian cycles Problem. | | | | | |
| **Unit** | **Module** | | **Micro content** | **No of Hrs** | |
| **IV** | Backtracking | | The General Method | 1 | |
| The N-Queens Problem - Description, State Space Tree, Algorithm. | 2 | |
| Sum of Subsets Problem - Description, Example, State Space Tree, Algorithm | 2 | |
| Graph Coloring Problem - Description, Example, State Space Tree, Algorithm. | 3 | |
| Hamiltonian Cycles Problem - Description, Example, State Space Tree, Algorithm. | 2 | |
| **UNIT V: 10 Hrs**  **Branch and Bound:** The General Method, FIFO Branch-and-Bound, LC Branch-and-Bound, 0/1 Knapsack Problem, Traveling Salesperson Problem. NP-Hard and NP-Complete problems: Basic concepts, Cook’s Theorem. | | | | | |
| **Unit** | **Module** | | **Micro content** | **No of Hrs** | |
| **V** | Branch and bound | | The General Method | 1 | |
| FIFO Branch and Bound | 1 | |
| LC Branch and Bound | 2 | |
| 0/1 Knapsack Problem - Description, Example | 2 | |
| Traveling Salesperson Problem - Description, Example | 2 | |
| NP-Hard and NP Complete problems | | Basics Concepts | 1 | |
| Cook’s Theorem | 1 | |

Code No :

**R20**

**III B. TECH II SEMESTER REGULAR EXAMINATION MODEL PAPER**

**DESIGN AND ANALYSIS OF ALGORITHMS**

**(CSM)**

**Time : 3 Hours Max. Marks : 70**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Note :** Answer **ONE** question from each unit **(5 × 14 = 70 Marks)**

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UNIT-I | | | | CO | BL |
| 1. | a) | Explain performance Analysis of algorithm. | [7M] | CO1 | L2 |
| b) | Explain Big-O Notation, Thetha Notation, Omega Notation with an example. | [7M] | CO1 | L2 |
| (OR) | | | |  |  |
| 2. | a) | What is Searching? Summarize different types of Searching with examples. | [7M] | CO1 | L2 |
| b) | What is Quick sort? Write an algorithm for Quick sort and find the time complixity.Illustrate the process of Quick Sort for elements 9,7,8,3,2,1. | [7M] | CO1 | L3 |
| UNIT-II | | | |  |  |
| 3. | a) | Explain the knapsack problem. Assume that we have a knapsack with max weight capacity, W = 16. Our objective is to fill the knapsack with items such that the benefit (value or profit) is maximum. Consider the following items and their associated weight and value   |  |  |  | | --- | --- | --- | | ITEM | WEIGHT | VALUE | | i1 | 6 | 6 | | i2 | 10 | 2 | | i3 | 3 | 1 | | i4 | 5 | 8 | | i5 | 1 | 3 | | i6 | 3 | 5 | | [7M] | CO2 | L3 |
| b) | Discuss the Dijkstra’s single source shortest path algorithm and derive its time complexity. Find shortest paths in the following graph using Dijkstra’s algorithm? | [7M] | CO2 | L3 |
| (OR) | | | |  |  |
| 4. | a) | Explain the Optimal storage on Tapes problem. Consider n=3 and lengths are (l1,l2,l3)=(5,10,3) find the optimal storage order. | [7M] | CO2 | L2 |
| b) | Explain the Optimal Merge Patterns Problem. Find optimal merge pattern for merging the following files  F1 with 13 values  F2 with 3 values  F3 with 9 values  F4 with 15 values  F5 with 8 values  F6 with 2 values | [7M] | CO2 | L2 |
| UNIT-III | | | |  |  |
| 5. | a) | Explain the All pairs shortest paths problem.  Find out the all pairs shortest paths for the given graph. | [7M] | CO3 | L2 |
| b) | Demonstrate pros and cons of 0/1 knapsack problem.  Find the optimal solution for the 0/1 knapsack problem making use of dynamic programming approach. Consider n = 4,w = 5 kg,(w1, w2, w3, w4) = (2, 3, 4, 5),(p1, p2, p3, p4) = (3, 4, 5, 6) | [7M] | CO3 | L2 |
| (OR) | | | |  |  |
| 6. | a) | Discuss the single-source shortest paths algorithm with a suitable example. Find shortest paths in the following weighted graph using Bellman Ford algorithm? | [7M] | CO3 | L2 |
| b) | Explain the travelling sales person problem. Find the shortest path by using travelling sales person problem for the given matrix. | [7M] | CO3 | L3 |
| UNIT-IV | | | |  |  |
| 7. | a) | Explain the Graph–coloring problem. And draw the state space tree for m= 3 colors n=4 vertices graph. Discuss the time and space complexity. | [7M] | CO4 | L3 |
| b) | Give difference between knapsack and 0/1 knapsack problem. | [7M] | CO4 | L2 |
| (OR) | | | |  |  |
| 8. | a) | Write an algorithm to determine the Hamiltonian Cycle in a given graph using backtracking. Find out the Hamiltonian paths for the given Graph. | [7M] | CO4 | L3 |
| b) | Explain the sum of subsets problem. Consider n=7,(w1,w2,w3,w4,w5,w6,w7)=(1,2,3,4,5,6,7) and sum m=10.find out the sum of subsets. | [7M] | CO4 | L3 |
| UNIT-V | | | |  |  |
| 9. | a) | Describe the Travelling sales person problem and discuss how to solve it using branch and bound. | [7M] | CO5 | L3 |
| b) | Develop an algorithm for 0/1 knapsack and write its advantages and dis-advantages with time complexity. | [7M] | CO5 | L3 |
| (OR) | | | |  |  |
| 10. | a) | Categorize the NP Problem. Explain NP-Hard and NP-Complete problem. | [7M] | CO5 | L4 |
| b) | Find the optimal solution using 0/1 knapsack problem using FIFO Branch and Bound. consider  n=4,(p1,p2,p3,p4)=(10,10,12,12,8),(w1,w2,w3,w4,w5)=(2,4,6,9) and m=15 find the optimal solution. | [7M] | CO5 | L3 |

**\* \* \* \* \***

**THE ABOVE MODEL PAPER ATTAINMENTS OF BLOOM’S TEXONOMY AS FOLLOWS**

**L1: 7\*7 = 49 = 35%**

**L2: 6\*7 = 42 = 30%**

**L3: 5\*7 = 35 = 25%**

**L4: 2\*7 = 14 = 10%**

SIGNATURES OF

COURSE COORDINATER MODULE COORDINATER HOD