MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu.

Department of Computer Science and Engineering
Question Bank - Academic Year (2021-22)

Course Code & Course Name : 19CSC11 & DESIGN AND ANALYSIS OF ALGORITHM

Name of the Faculty : R.NIVETHITHA

Year/Sem/Sec : II/IV/C

Unit-I: Introduction
Part-A (2 Marks)

1. Define Algorithm & its features.
2. What is Algorithm Design Technique or algorithmic strategy?
3. How to measure an algorithm's running time?
4. Define the asymptotic notation & its types.
5. Define the Basic Operation.
6. Explain Euclid's algorithm.
7. State the different types of Problems.
8. State the two kinds of algorithm efficiency.
9. State the Order of growth for an Algorithm & Asymptotic Notations.
10. How algorithms efficiency is measured?
11. Explain the methods of specifying an algorithm.
12. Explain various basic efficiency classes.
13. Compare Orders of growth of n (n-1)/2 & n^2
14. Compare Orders of growth of n! and 2^n
15. What are drawbacks in using standard unit of time to measure runtime of algorithm.
16. What is average case analysis?
17. What is binary search?
18. List out two drawbacks of binary search algorithm
19. Give the recurrence relation for the worst case behavior of merge sort.
20. Define Pivot element.
21. Define Merge Sort.
22. Is Quick sort stable sorting algorithm?
23. State Divide and Conquer Technique.

24. State the worst, best, average case Efficiency of Quick Sort and Merge sort.

25. Write an Algorithm for Quick sort.

Part-B (16 Marks)

1. Describe the steps in analyzing & coding an algorithm.
2. Explain the various asymptotic notations used in algorithm design.
3. Explain the general framework for analyzing the efficiency of algorithm.
4. Explain in detail about fundamentals of algorithmic problem solving with example.
5. Explain in detail about mathematical analysis of recursive algorithms with suitable problem.
6. Explain in detail about mathematical analysis of non-recursive algorithms.
7. Explain the recursive algorithm for computing the Fibonacci series and analyze it.
8. Explain the towers of Hanoi problem and solve it using Recursion.
9. Explain in detail about merge sort. Illustrate the algorithm with numeric example. Provide the complete analysis for the same. 286, 45,278,368,475,389,656,788,503,126
10. Sort the following elements using merge sort 12, 24,8,71,4,23,6,89,56
11. Using Divide & Conquer Explain Binary search algorithm in detail with example

Unit-II : BACKTRACKING

Part-A (2 Marks)

1. Define backtracking with example.
2. Explain the Brute Force approach
3. List out disjoint set algorithms.
4. Define spanning tree.
5. What is exhaustive search?
6. Define quick hull.
7. What is Convex hull problem?
8. Define Knapsack's problem.
9. State the problem of sum of subset.
10. State the formula for Strassen's Matrix Multiplication.
11. Define Assignment problem.
12. Define Traveling Salesman problem.
13. What is Hamiltonian circuit?
14. Define Dynamic programming.
15. Define N queen problem.
16. Define graph coloring.

Part-B (16 Marks)

1. Write exhaustive search algorithm for traveling salesman, Knapsack and Assignment problem.
2. Explain in detail about N queen problem using backtracking discuss the possible solutions
3. Explain the method of multiplication of large numbers with the help of illustrative example
4. Write down the steps involved to find sum of subsets using backtracking method with an example.
5. Explain the 4-queen's problem using backtracking. Write the algorithms. Give the estimated cost for all possible solutions of 4-queen's problem. Specify the implicit and explicit constraint.
6. Apply backtracking technique to solve the following instance of subset sum problem: S={1,3,4,5} and d=11 ii. s=(1,3,4,5) & d=11

Unit-III : GREEDY METHOD

Part-A (2 Marks)

1. Define Greedy method.
2. Explain types of Tree Traversals.
3. Define minimum spanning tree with example.
4. Define Kruskals Algorithm.
5. Write the Algorithm for Prims algorithm.
6. What are the problems that can be solved using Greedy technique.
7. Compare Greedy Technique& Dynamic Programming.
8. Define all-pairs shortest-path problem
9. Define Di-graph with example.
10. Define Weighted graph with example.
11. Define Adjacency matrix.
12. Define Transitive closure.
13. Define Warshall's algorithm
14. Define Floyds algorithm.
15. Define Distance Matrix.
16. Define Principle of Optimality.
17. State the formulas of Floyds algorithm & Warshalls algorithm.
18. Difference between Dynamic programming & Divide-Conquer.
19. State the algorithm for memory function.

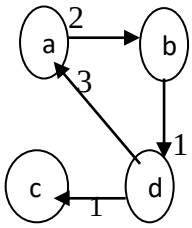
20. Define single source shortest path method.

21. Define Optimal Binary Search tree.

22. Compare feasible and optimal solution

23. What is an optimal solution?

24. Find the minimum spanning tree for the given graph:



Part-B (16 Marks)

1. Using OBST algorithm (a1,a2,a3,a4) = (end, goto, print, stop) with

   P_1=1/20, P_2=1/5, P_3=1/10, P_4=1/20

   q_o=1/5, q_1=1/10,q_2=1/5,q_3=1/20, q_4=1/20

   construct the optimal binary search tree.

2. Describe in detail about prim's algorithm with suitable example.
3. Explain in detail about kruskal's algorithm.
4. Solve the all-pairs shorest path problem for the digraph with the weight matrix given below

   A B C D

   A 0 ∞ ∞ 3

   B 2 0 ∞ ∞

   C ∞ 7 0 1

   D 6 ∞ ∞ 0

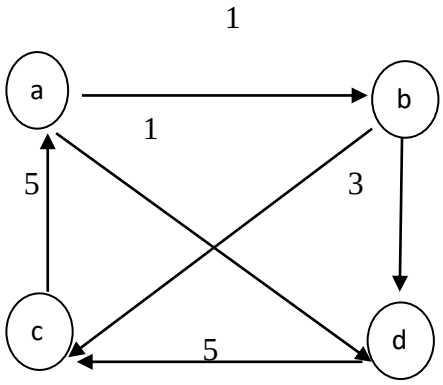
5. Find the optimal solution to the fractional knapsack problem with given data:

Item	Weight	Benefit
A	2	60
B	3	75
C	4	90

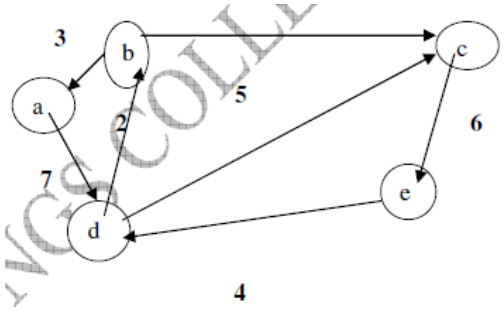
6. There are 4 people who need to be assigned to execute 4 jobs(one person per job) and the problem is to find an assignment with the minimum total cost. The assignment costs is given below, solve the assignment problem by exchaustive search. (nov/dec 2016)

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

7. Explain Wars hall's & Floyd's Algorithm with example .Apply Floyd's algorithm or obtain all pair shortest path for the following graph. Explain with the algorithm.



8. Solve the following instance of the single source shortest path problem with vertex 'a' as the source using Dijkstra's algorithm. May/Jun 2014



Unit-IV : DYNAMIC PROGRAMMING

Part-A (2 Marks)

1. Define Iterative Improvement technique.
2. Define Optimal solution.
3. Define Simplex Method.
4. Define Feasible Solution
5. What are the requirements of standard form?
6. What are the steps involved in simplex method?
7. Define Sink, Source and Capacity.
8. Define matching.
9. Define bipartite graph.
10. Define maximum matching or maximum cardinality matching
11. What is stable marriage problem?
12. Define flow network.
13. What is maximum flow problem?

14. Define forward and backward edges.

15. Define shortest-argument-path or first-labled-first scanned algorithm

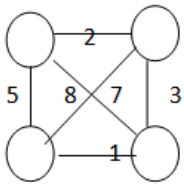
16. Define shortest-augmenting-path or first-labeled-first scanned algorithm

17. What is two colorable graph?

18. What is Linear Programming problem?

Part-B (16 Marks)

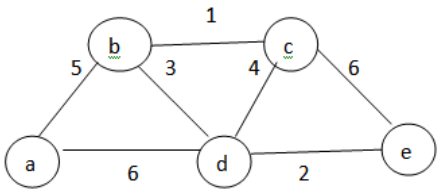
1. Fine all solution to the travelling salesman problem (cities and distances shown below) by exhaustive search. Give the optimal solution.



2. Solve the all-pairs shortest path problem for the digraph with the following weight matrix:

0	2	∞	1	8
6	0	3	2	∞
∞	∞	0	4	∞
∞	∞	2	0	3
3	∞	∞	∞	0

3.Apply Kruskal's algortihm to find a minimum spanning tree of the following graph.



4. Solve the following instance of the 0 / 1. Knapsack problem give the knapsack capcity in W=5 using dynamic programming and explain it.

Item	Weight	Value
1	4	10
2	3	20
3	2	15
4	5	25

5. Apply the dynamic programming following instance of the knapsack problem and solve

Item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

Unit-V : BRANCH AND BOUND&NP-HARD, NP-COMPLETE PROBLEMS

Part-A (2 Marks)

1. Define Decision trees.
2. In Backtracking method, how the problem can be categorized?
3. What are the two types of constraints in backtracking?
4. Define State space tree.
5. Define Promising node
6. Define Non-Promising nodes
7. Define Subset Sum problem.
8. Define Hamilton Circuit problem
9. Define n-Queens problem.

Part-B (16 Marks)

1. Explain how branch and bound technique is used to solve knapsack problem.
   for n=4 , W=10, $(p_1,p_2,p_3,p_4) = (40,42,25,12)$ and $(w_1,w_2,w_3,w_4) = (4,7,5,3)$.
2. Explain about assignment problem using branch and bound with example.
3. Discuss the solution for travelling salesman problem using branch & bound technique.
4. Discuss the decision trees for sorting algorithms.
5. Solve the following instance of the Knapsack problem for the given knapsack capacity M=4 using branch and bound algorithm

Item	Weight	Value
1	2	12
2	1	10
3	3	20
4	2	15

6. Implement an algorithm for knapsack problem using NP-Hard approach

.

Course Faculty **HoD**