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 Information Technology
Question Bank - Academic Year (2020-21)

Course Code & Course Name	**:**	**19ITC11 & DESIGN AND ANALYSIS OF ALGORITHM**
Name of the Faculty	**:**	**T.Manivel**
Year/Sem/Sec	**:**	**II/IV/-**

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

1. Define Algorithm & its features.
2. What is Algorithm Design Technique or algorithmic stategy?
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. Write a recursive algorithm to find number of binary digits in binary representation of an integer.
16. Write a recursive algorithm for computing FACTORIAL of a number?
17. Write a algorithm to find element uniqueness in an array.
18. Write a recursive algorithm for computing the nth Fibonacci number?
19. State the following Terms: Time Complexity & Space Complexity.
20. What is Exact and Approximation algorithm?
21. What do you mean by "Worst case-Efficiency" of an algorithm?
22. Define Big omega notation.
23. Define Recurrence Equation

24. What are drawbacks in using standard unit of time to measure runtime of algorithm.

25. What is average case analysis?

26. Define program proving and program verification.

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 proble
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.

Unit-II : Brute Force and Divide-and-Conquer

Part-A (2 Marks)

1. Explain the Brute Force approach
2. What is exhaustive search?
3. Define quick hull.
4. What is Convex hull problem?
5. What is binary search?
6. What is the difference between Merge sort and Quick sort.
7. List out two drawbacks of binary search algorithm
8. Give the recurrence relation for the worst case behavior of merge sort.
9. Define Pivot element.
10. Define Merge Sort.
11. Is Quick sort stable sorting algorithm?
12. State Divide and Conquer Technique.
13. State the worst, best, average case Efficiency of Quick Sort.
14. State the worst, best, average case Efficiency of Merge sort.
15. What are the problems that can be solved using Divide & Conquer method?
16. Write an Algorithm for Quick sort.
17. Define Knapsack’s problem.
18. State the formula for Strassen’s Matrix Multiplication.
19. Define Assignment problem.
20. Define Traveling Salesman problem.
21. What is Difference between quick sort and merge sort?

22. Give the Time efficiency and drawback of merge sort algorithm

Part-B (16 Marks)

1. 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**
2. Sort the following elements using merge sort **12, 24,8,71,4,23,6,89,56**
3. Write exhaustive search algorithm for traveling salesman, Knapsack and Assignment problem.
4. Using Divide & Conquer Explain Binary search algorithm in detail with example.
5. Explain the method of multiplication of large numbers with the help of illustrative example
6. Explain the Quick sort algorithm with the help of illustrative example?
7. Explain closest pair algorithm & upper and lower hulls in the convex-hull problem, with an example.
8. Differentiate sequential search from binary search Technique

Unit-III : Dynamic Programming and Greedy Technique

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 Dynamic programming.
9. Define all-pairs shortest-path problem
10. Define Di-graph with example.
11. Define Weighted graph with example.
12. Define Adjacency matrix.
13. Define Transitive closure.
14. Define Warshall’s algorithm
15. Define Floyds algorithm.
16. Define Distance Matrix.
17. Define Principle of Optimality.

18. State the formulas of Floyds algorithm & Warshalls algorithm.

19. Difference between Dynamic programming & Divide-Conquer.

20. State the algorithm for memory function.

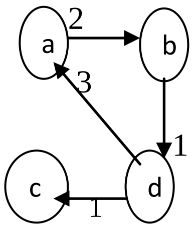
21. Define single source shortest path method.

22. Define Optimal Binary Search tree.

23. Compare feasible and optimal solution

24. What is an optimal solution?

25. 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

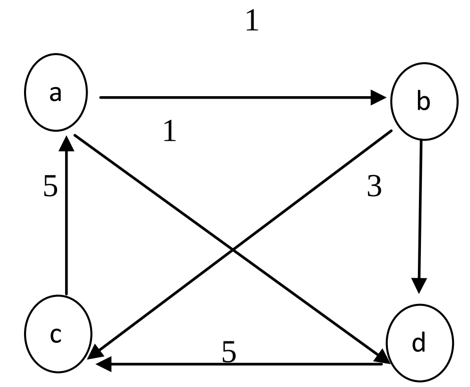
   P_1=1/20, P_2=1/5, P_3=1/10, P_4=1/20

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

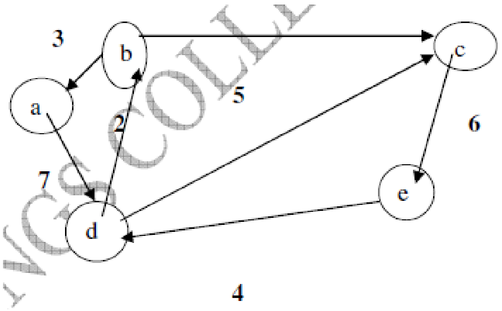
   construct the optimal binary search tree.
2. Describe in detail about prim's algorithm with suitable example.
3. Explain in detail about kruskal's algorithm.
4. Solve the all-pairs shorest path problem for the digraph with the weight matrix given below

$$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & \infty & \infty & 3 \\ B & 2 & 0 & \infty & \infty \\ C & \infty & 7 & 0 & 1 \\ D & 6 & \infty & \infty & 0 \end{array}$$

5. 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.



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



7. 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

8. Explain in detail Huffman Trees with suitable example.

Unit-IV : Iterative Improvement and Limitation of Algorithm

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 augmenting-path method or Ford-Fulkerson method

15. Define forward and backward edges

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

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

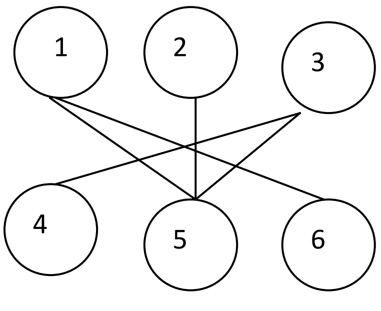
18. What is two colorable graph?

19. What is entering variable?

20. What is Linear Programming problem?

Part-B (16 Marks)

1. Explain in detail about an outline of an simplex method with example.
2. Write short notes on the following:
   i) Flow conservation requirement
   ii) Augmenting path method
3. Discuss about the Following:
   i) Shortest augmenting path algorithm
   ii) Forward and Backward edges
4. Explain how the maximum flow problem for a network with several sources and sinks can be transformed into the same problem for a network with a single source and a single link.
5. Write an algorithm for Maximum Bipartite matching with example.
6. Write an algorithm for stable marriage algorithm with example.
7. Apply the maximum matching algorithm to following bi-partite graph



8. Consider that there are 4 men and 4 women. The men are named as A,B,C,D and Women are named

as X,Y,Z,W. Preference list is given. Find the matching pair which is stable.

Mens preferences

	1st	**2nd**	**3^{rd}**	**4th**
A	Y	W	X	Z
B	W	X	Y	Z
C	Y	Z	W	X
D	Y	Z	W	X

	1st	**2nd**	**3rd**	**4th**
X	B	C	D	A
Y	C	D	B	A
Z	C	B	A	D
W	C	B	D	A

Unit-V : Backtracking ,Branch and Bound and Approximation Algorithm

Part-A (2 Marks)

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

Part-B (16 Marks)

1. Explain the N-Queen's problem & discuss the possible solutions. (Nov/Dec 2011), May/June 12
2. Apply backtracking technique to solve the following instance of subset sum problem: (Nov/Dec 2010)

i. S={1,3,4,5} and d=11 ii. s=(1,3,4,5) & d=11 (16)

3. Explain how branch and bound technique is used to solve knapsack problem. Apr-May 2010

   for n=4 , W=10, $(p_1,p_2,p_3,p_4) = (40,42,25,12)$ and $(w_1,w_2,w_3,w_4) = (4,7,5,3)$.
4. Explain about assignment problem using branch and bound with example.
5. Discuss the solution for travelling salesman problem using branch & bound technique. Apr-May 2010
6. Discuss the decision trees for sorting algorithms.
7. Write the Approximation algorithms of the knapsack & traveling salesman problem.
8. Solve the following instance of the Knapsack problem for the given knapsack capacity M=4 using

   branch and bound algorithm. (Nov/Dec 2010), Apr-May 2010

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

Course Faculty **HoD**