SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING AND TECHNOLOGY
MUKKA, MANGALURU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

DESIGN AND ANALYSI S OF ALGORITHMS

SUBJECT CODE: 19SCS42

COMPILED BY:

Ms. Swarna H.R., Assistant Professor

MODULE 1
INTRODUCTION TO ALGORITHMS

Sl. No.	Questions
1	Define algorithm. Discuss the criteria's that an algorithm must satisfy with an example.
2	Define best case, worst case and average case efficiency. Write the algorithm and give these efficiencies for sequential search.
3	Explain space complexity and time complexity with an example.
4	Explain with an example how a new variable count introduced in a program can be used to find the number of steps needed by a program to solve a particular problem instance.
5	Consider the following algorithm. Algorithm GUESS (A[ ] [ ]) for i ← 0 to n – 1 for j ← 0 to i A [i] [j] ← 0 i) What does the algorithm compute? ii) What is basic operation? iii) What is the efficiency of this algorithm?
6	Explain asymptotic notations Big O, Big Ω and Big θ that are used to compare the order of growth of an algorithm with example.
7	Describe various basic efficiency classes.
8	Prove the following statements. a. $n^2 + 5n + 7 = \Theta(n^2)$ b. ½ n(n-1) = $\Theta(n^2)$ c. ½ n^2+3n= $\Theta(n^2)$ d. 100n + 5 =`O(n2) e. $n^2 + n = O(n^3)$ f. $5n^2 + 3n + 20 = O(n^2)$ g. $n^3 + 4n^2 = \Omega(n^2)$
9	Define Little Oh**.** Compare the orders of growth of following functions i) (½) n (n-1) and n^2 ii) 3n+2 and n^2
10	Explain general plan of mathematical analysis of **non-recursive** algorithms with example.
11	Write the algorithm to find **maximum element** in the given array and explain the mathematical analysis of this non-recursive algorithm.

12 Write the algorithm to check whether all the elements in the given array are **distinct** and explain the mathematical analysis of this non- recursive algorithm. Derive its worst-case time complexity

13 Write the algorithm to perform **matrix multiplication** and explain the mathematical analysis of this non-recursive algorithm

14 Explain general plan of mathematical analysis of **recursive algorithms**

with example.

15 Illustrate mathematical analysis of recursive algorithm for **Towers of Hanoi** OR

Give the recursive algorithm to solve Tower of Hanoi problem. Show that the efficiency of this algorithm is exponential

16 Illustrate mathematical analysis of recursive algorithm to find the

factorial of a given number.

17 State the recursive algorithm to count the **bits of a decimal number** in its binary representation. Give its mathematical analysis.

18 Write a recursive function to find and print all possible **permutations**

of a given set of n elements

19 Solve the recurrence relation $M(n) = 2M(n-1) +1$ for $n>1$; $M(1)=1$

20 Briefly explain the **important problem types** coming under design and analysis of algorithms.

21 Explain the following types of problems:

i) Combinatorial problems ii) Graph problems

22 Briefly explain important **fundamental data structures** used in algorithm design.

23 Explain two common ways to represent the **graph** with example

24 Discuss adjacency matrix and adjacency list representation of a graph with suitable example.

MODULE 2
DIVIDE AND CONQUER

Sl. No.	Questions
1	Explain the general concept of divide and conquer method. Give the general algorithm DAndC(P) [Where P is the problem to solve] to illustrate this technique. Discuss general divide and conquer technique with control abstraction and recurrence relation.
2	List the advantages and disadvantages of Divide and Conquer.
3	Find the upper bound of recurrences given below by substitution method. i) $T(n) = 2\,T(n/2)+n$ ii) $T(n) = T(n/2) + 1$
4	Explain the general method of substitution method to solve the recurrence equation.
5	State and explain master theorem to solve the recurrence equation.
6	Consider Tower of Hanoi puzzle. Derive the recurrence relation for the total movement of disk. Solve the recurrence relation using substitution method
7	Show how **binary search** problem can be solved using the divide and conquer method. Write an algorithm for binary search and find average case efficiency.
8	Design an algorithm to find the **maximum and minimum** element in a given list of n numbers using divide and conquer method.
9	Write the algorithm for **Merge Sort**. Illustrate with an example. Derive the time efficiency (best case, average case, worst case) of the algorithm.

10 Sort the following elements using **merge sort.** Write the recursion tree. 70, 20, 30, 40, 10, 50, 60

Alternatively: Use D & C method to sort the numbers which divides problem size by considering position

11 Write algorithm for Merge Sort and Trace 60, 50, 25, 10, 35, 25, 75, 30.

12 Write an algorithm for sorting the numbers using **Quick sort**. Derive the best case, worst case, average case time efficiency of the algorithm.

13 Discuss how quick-sort works to sort an array and trace for the following dataset. Draw the tree of recursive calls made.

65, 70, 75, 80, 85,60, 55, 50, 45

14 Illustrate the tracing of quick sort algorithm for the following set of numbers: 25, 10, 72, l8, 40, 11, 64, 58, 32, 9

15 Sort the list E, X, A, M, P, L, E in alphabetical order using the Quick Sort algorithm. Draw the tree of recursive call.

16 Trace the quicksort algorithm for following list in ascending order 80, 60, 70, 40, 10, 30, 50, 20

17 Explain the algorithm which is used to perform matrix multiplication in an efficient way. OR

Briefly explain Strassen's matrix multiplication and how its uses divide and conquer method. Obtain the time complexity.

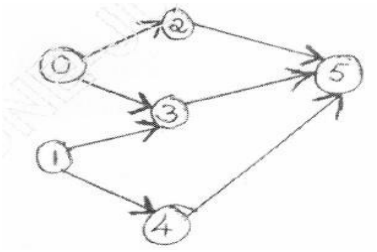
18 Apply Strassen's matrix multiplication to multiply following matrices. Discussmethod is better than directmatrix multiplication method.

$$\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 5 \\ 1 & 6 \end{bmatrix}$$

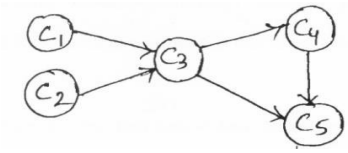
19 What are the three major variations of decrease and conquer technique? Explain with an example for each.

20 Explain topological sorting with example.

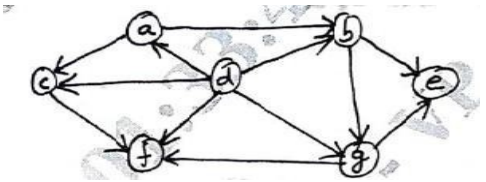
21 Apply source removal method to obtain topological sort for the given graph.



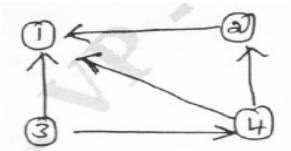
22 Illustrate the topological sorting for the following graph.



23 Apply DFS for below graph to solve to topological sorting.



24 Apply DFS for below graph to solve to topological sorting.



MODULE 3

GREEDY METHOD

1 Define Optimal solution and feasible solution.

2 Define coin change Problem. State the greedy method to solve the coin change problem. For 49 rupees, find the denominations with least no. of coins. The available denominations in rupees are { 1, 2, 5, 10}

3 Define coin change problem. If coins available are of values { 2, 5, 3, 6 }, find the least denominations for a) 55 b)77

4 What is job sequencing with deadline problem.

5 Obtain the optimal solution for the job sequencing problem with deadline where n = 4, (P_1, P_2, P_3, P_4) = (100, 10, 15, 27), (d1, d2, d3, d4) = (2, 1, 2, 1)

6 What is the solution generated by job sequencing when n = 5, (P_1, P_2, P_3, P_4, P_5) = (20, 15, 10, 5, 1), (d1, d2, d3, d4, d5) = (2, 2, 1, 3, 3)

7 Find solution generated by job sequencing problem with deadlines for 7 jobs given profits 3, 5, 20, 18, 1, 6, 30 and deadlines 1, 3, 4, 3, 2, 1, 2 respectively.

8 Let n = 5, profits [10, 3, 33, 11, 40] and deadlines [3, 1, 1, 2, 2] respectively. Find the optimal solution using greedy algorithm.

9 What is knapsack problem?

10 Obtain solution for a knapsack problem using greedy method for n = 3, capacity m=20 values 25, 24, 15 and weights 18, 15, 10 respectively.

11 Apply greedy method to obtain an optimal solution to the knapsack problem given M = 60, (w1, w2, w3, w4, w5) = (5, 10, 20, 30, 40), (p1,

p2, p3, p4, p5) = (30, 20, 100, 90, 160). Find the total profit earned.

12 Solve the greedy knapsack problem where m=10, n=4, P=(40, 42, 25, 12), W=(4, 7, 5, 3)

13 Find the optimal solution for the knapsack problem where n=7, m=15 using greedy method

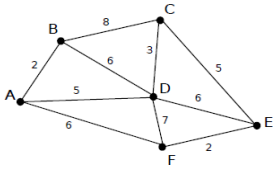
Object	1	2	3	4	5	6	7
Weight	02	03	05	07	01	04	01
Profit	10	05	15	07	06	18	03

14 Define MST. Write Prim's algorithm to construct minimum cost spanning tree.

15 Write Kruskals algorithm to construct MST. Show that the time efficiency is O(|E|log|E|)

16 Apply Prims and Kruskal algorithm to find the MST. Show the intermediatesteps.

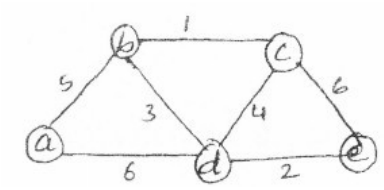
17 Apply PRIMS and KRUSKAL algorithm for the following graph to get MST. Show the intermediate steps.

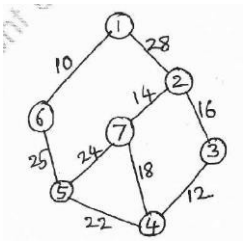


18 Obtain minimum cost spanning tree for the graph whose weight matrix is given below.

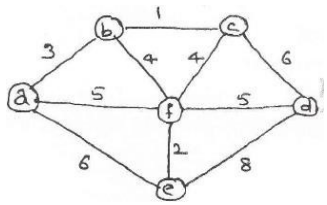
$$\begin{bmatrix} 0 & 3 & \infty & 7 & \infty \\ 3 & 0 & 4 & 2 & \infty \\ \infty & 4 & 0 & 5 & 6 \\ 7 & 2 & 5 & 0 & 4 \\ \infty & \infty & 6 & 4 & 0 \end{bmatrix}$$

19 Apply Prims and Kruskal algorithm to find the MST. Show the intermediate steps.



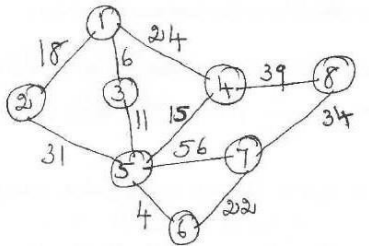


20 Apply Prims and Kruskals algorithm to find the MST of the graph given below.

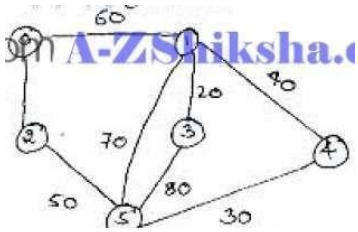


21 Apply Prim's algorithm to obtain a minimum spanning tree for the given weighted connected graph.

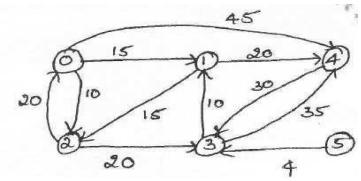
a)



b)

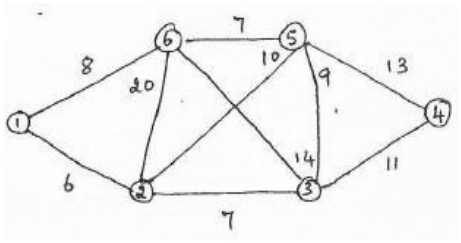


22 Apply Dijikstra's algorithm to find single source shortest path for the graph given below. Source vertex is 5.

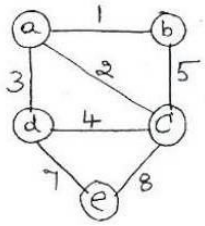


23 Apply Prim's algorithm to obtain a minimum spanning tree for the given weighted connected graph.

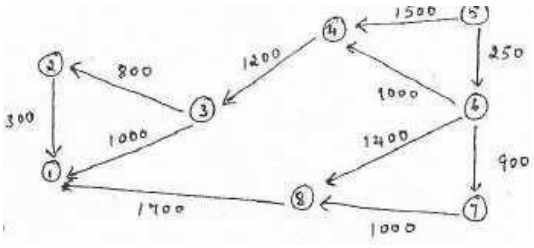
a)



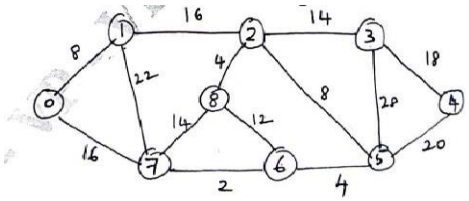
b)



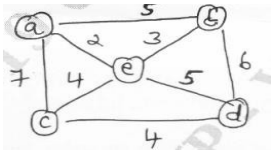
c)



24 Find the MST using Kruskal's Algorithm

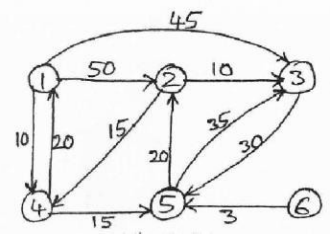


24 Find the MST using Kruskal's Algorithm

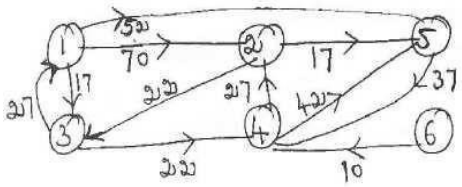


25 Write Dijikstra's algorithm to find single source shortest path OR Write an algorithm to find single source shortest path

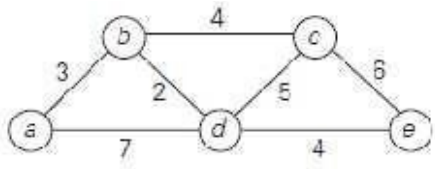
26 Apply Dijikstra's algorithm to find single source shortest path for the graph given below. Consider Node 6 as source.



27 Determine the shortest paths from vertex 1 to all other vertices.



28 Apply single source shortest path finding algorithm. Consider source vertex as (a)



29 Explain Huffman coding algorithm. With an example show the construction of Huffman tree and generate the Huffman code

30 Explain Bottom-up heap construction algorithm with an example. Give the worst-case efficiency.

31 Construct a heap for the list 1, 8, 6, 5, 3, 7, 4 by the bottom-up algorithm.

32 Sort the following lists by heapsort by using the array representation of heaps.5, 2, 4, 1, 3 (in increasing order)

33 Sort the array 2, 9, 7, 6, 5, 8 by heapsort.

MODULE 5
BACKTRACKING

1 What is backtracking. Give the general Procedure. OR Write the pseudocode for backtracking algorithm

2 Give the problem statement of n-queens problem. Explain the solution for 4-queens problem using state space tree.

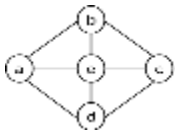
3 Write a note on Sum of Subset problem

4 Let w = {3, 5, 6, 7} and m =15. Find all possible subsets of w that sum to m. Draw the state space tree that is generated

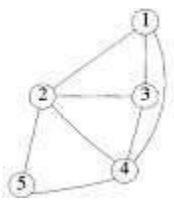
5 Apply backtracking to solve the following instance of the subset-sum problem : S={1,3,4,5} and d=11. Draw the state space tree.

6 Apply backtracking to solve the following instance of the subset-sum problem : S={5,10,12,13,15,18} and d=30. Give all possible solutions.

7 Define Graph coloring problem. Apply backtracking to solve the 3-cloring problem for the graph given below.



8 Apply backtracking based graph coloring algorithm for the graph given below with m=4. Give state space tree showing first 3 valid assignments.

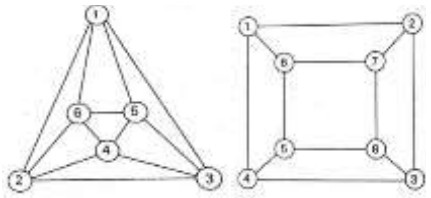


9 Draw the portion of the state space tree for m – colorings of a graph when n=4 and m=3

10 Find different solutions for 4 nodes and all possible3 coloring problem

11 What is Hamiltonian cycle? Give the backtracking based algorithm to find the Hamiltonian cycle in the graph. Write the functions used to generating next vertex and for finding Hamiltonian cycles.

12 Apply the backtracking to the problem of finding Hamiltonian cycle in the following graphs



(a) (b)

13 What branch and bound method. How it is different from backtracking.

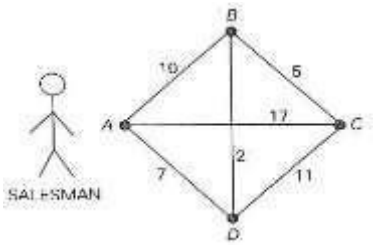
14 Apply best-first branch and bound method for the following instance of assignment problem to find the optimal solution. Give the complete state space tree.

Job 1	Job 2	Job 3	Job 4	
9	2	7	8	Person a
6	4	3	7	Person b
5	8	1	8	Person c
7	6	9	4	Person d

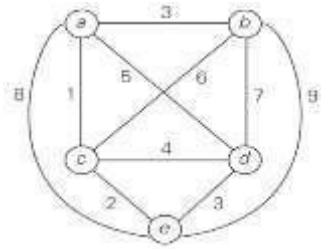
What are the two additional items required by branch and bound technique, compared with backtraking?

15 Explain how TSP can be solved using branch and bound technique.

16 Apply the branch-and-bound algorithm to solve the travelling sales man problemfor the following graph. Consider start city as A. Give the statespace tree.



17 Apply the branch-and-bound algorithm to solve the travelling sales man problemfor the following graph. Start city is ***a***. Give the statespace tree.



18 Explain LC branch & bound and FIFO branch and bound for knapsack problem

19 Write a note on deterministic and non deterministic algorithms.

20 Explain the following with examples

a. Class P Problems

b. Class NP Problems
c. NP complete problem
d. NP hard problem.

19 With the help of a state space tree, solve the following instance of Knapsack problem by the branch and bound algorithm. Knapsack Capacity W = 10

Item No.	1	2	3	4
Weight	4	7	5	3
Value	40	42	25	12

20 With the help of a state space tree, solve the following instance of Knapsack problem by the branch and bound algorithm. Knapsack Capacity W = 15

Item No.	1	2	3	4	5	6
Weight	5	7	2	4	5	1
Value	40	35	18	4	10	2

21 Apply **Least Cost Branch and Bound** (LCBB) method for the following instance of 0/1 Knapsack problem to get the optimal solution. Knapsack Capacity W = 15

Item No.	1	2	3	4
Weight	2	4	6	9
Value	10	10	12	18

22 Apply **FIFO Branch and Bound** method for the following instance of 0/1 Knapsack problem to get the optimal solution.Knapsack Capacity W = 15

Item No.	1	2	3	4
Weight	2	4	6	9
Value	10	10	12	18