SRM VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

IV SEMESTER

1904402 - DESIGN AND ANALYSIS OF ALGORITHMS

Regulation – 2019

Academic Year 2022 – 23

Prepared by

Dr. B. Muthu Senthil, Professor/CSE

Dr. V. Dhanakoti , Professor/CSE

Dr. L. Karthikeyan , Assistant Professor (Sel.Grade) / CSE

SRM VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QUESTION BANK

SUBJECT CODE/NAME: 1904402 DESIGN AND ANALYSIS OF ALGORITHMS

SEM / YEAR: IV/II

UNIT I - INTRODUCTION			
Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and their properties. Analysis Framework – Empirical analysis - Mathematical analysis for Recursive and Non-recursive algorithms – Visualization			
PART – A			
Q. No	**Questions**	**BT Level**	**Competence**
1.	**Define** algorithm and list the desirable properties of an algorithm.	Remember	BTL-1
2.	**Define** best, worst and average time complexity.	Remember	BTL-1
3.	**How** to measure the algorithm running time?	Understand	BTL-2
4.	**List the steps** to write an Algorithm.	Analyze	BTL-4
5.	**Prove** that if f(n)=O(g(n)) ad g(n) = O(f(n)), then f(n)=Θ g(n).	Apply	BTL-6
6.	**Evaluate** an algorithm for computing gcd(m,n) using Euclid's Algorithm	Evaluate	BTL-5
7.	**Design** the equality gcd(m,n)=gcd(n,m mod n) for every pair of positive integers m and n.	Create	BTL-6
8.	**What** do you mean by the **"Worst Case efficiency**" of an algorithm?	Remember	BTL-1
9.	**Identify how** you will measure input size of algorithms.	Remember	BTL-1
10.	**Explain** the various types of problems that can be solved using algorithm.	Analyze	BTL-4
11.	**Apply** the common technique for proving the correctness of an algorithm.	Apply	BTL-3
12.	**How** will you measure the total running time for a problem?		
13.	**List** the most important problem types.	Remember	BTL-3
14.	**List** the application of Graph problems..	Remember	BTL-3

15.	**Define** Big 'Theta' notation.	Apply	BTL-1
16.	**Define** Big 'Oh' notation.	Remember	BTL-1
17.	**Define** Big 'Omega' notation.	Remember	BTL-1
18.	**Formulate** the order of growth. Compare the order of growth n! and 2^n.	Create	BTL-6
19.	**Differentiate** between Best, average and worst case efficiency.	Understand	BTL-2
20.	**Discuss** the concepts of asymptotic notations and its properties.	Understand	BTL-2
20.	**Analyze** the order of growth. (i).F(n) = $2n^2$ + 5 and g(n) = 7n. Use the Ω (g(n)) notation.	Analyze	BTL-4
21.	Differentiate Algorithm VS Program.	Evaluate	BTL-5
22.	Express the recurrence relation for recursive algorithm.	Evaluate	BTL-5
23.	**Discuss** the General plan for analyzing efficiency of Non recursive & Recursive algorithms.	Understand	BTL-2
24.	**Discuss** the following questions by consider the definition based algorithm for adding two n by n matrices. 1. What is basic operation? 2. How many times it is performed as a function of the matrix order n? 3. How many times it is performed as a function of the total number of elements in the input matrices?	Understand	BTL-2

	PART - B		
1.	**Give** the General Plan for Analyzing the Time Efficiency of Recursive Algorithms and use recurrence to find number of moves for Towers of Hanoi problem n	Remember	BTL-1

2.	(i) Consider the following algorithm for the searching problem. (8)
	ALGORITHM Linear search (A[0,…….n-1],key)
	// Searches an array for a key value by Linear search.
	//Input: Array A [0…..n-1] of values and a key value to search.
	//Output: Returns index if search is successful.
	For i← 0 to n-1 do
	If [key== A[i])
	Return i.
	a) **Apply** this algorithm to search the list 10, 92,38,74,56,19,82,37 for a key value 74.
	b) Is this algorithm efficient?
	c) When can this algorithm be used?
		Apply	BTL-3

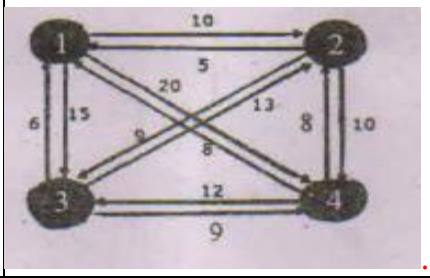
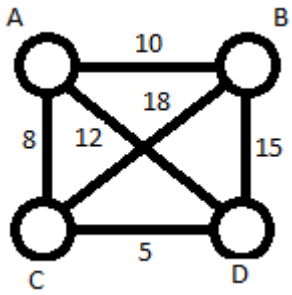
	(ii) What are the most important problem types are used to illustrate different algorithm design techniques and methods of algorithm analysis. (5)		

3.	Write the asymptotic notations used for best case, worst case and average case of algorithms. (7) Create an algorithm to find the maximum element in an array. Give best, worst and average case complexities. (6)	Create	BTL-6
4.	For each of the following algorithms, i) Compute n! (7) **ii)Asses & find** the largest element in a list of n numbers with respect to the following conditions: (6) (a). A natural size metric for its inputs. (b). Its basic operation. (c). Whether the basic operation count can be different for inputs of the same sizes.	Analyze	BTL-5
5.	**Discuss** various methods used for mathematic analysis for recursive and non-recursive algorithms. (13)	Understand	BTL-2
6.	What are the rules of manipulate Big Oh expressions and about the typical growth rates of algorithms? (13)	Understand	BTL-2
7.	Illustrate briefly on Big oh Notation, Omega Notation and Theta Notations. Depict the same graphically and explain. (13)	Evaluate	BTL-3
8.	**(i)Define a** Mathematical analysis of recursive algorithms. (4) (ii)**Examine** the efficiency of factorial of some number n with the help of General plan. (9)	Remember	BTL-1
9.	**(i)Define a** Mathematical analysis of Non-recursive algorithms. (5) (ii) **Tell** about the efficiency of finding the element with maximum value in a given Array with the help of General plan. (8)	Remember	BTL-1
10.	**(i)Define** Towers of Hanoi problem. (3) (ii)**Describe** the time complexity of Towers of Hanoi problem. (10)	Remember	BTL-1
11.	**Explain** in detail about Analysis Framework with a suitable Example (13)	Analyze	BTL-4
12.	Analyze the recursive and non-recursive versions of the factorial function. i) Examine how much each function requires as 'n' becomes large. (7) ii) Find the time complexity and space complexity (6)	Evaluate	BTL-3
13.	Evaluate the recurrence relations. (i). x (n) = x (n-1) + 5 for n>1. (7) (ii). X (n) = x(n/3) +1 for n >1,x(1) =1. (Solve for n = 3k) (6)	Apply	BTL-4
14.	Discuss in detail about the fundamentals of algorithmic problem solving. (13)	Understand	BTL-2
15.	**Examine** the recurrence relation efficiency of converting the bits in binary expansion. (13)	Analyze	BTL-5
16.	**Evaluate** the recurrence relation for finding fibonocci series. (13)	Apply	BTL-4
17.	**Explain** in detail basic asympototic efficeincy classes with its comments (13)	Understand	BTL-2

PART C			
1.	**Evaluate** the following equalities are correct: i)$5n^2-6n=\Theta(n^2)$ (4) ii)$n!=O(n^n)$ (4) iii)$n^3+10^6n^2=\Theta(n^3)$ (4) iv)$2n^2 2^n+n \log n=\Theta(n^2 2^n)$ (3)	Evaluate	BTL-5
2.	**Evaluate** the following recurrences completely i) $T(n) = T(n/2) + 1$, where $n=2^k$ for all $k \geq 0$ (4) ii) T(n) = T(n/3) + T(2n/3) + cn, where 'c' is a constant and 'n' is the input size. (4) iii) Explain the steps involved in problem solving (7)	Evaluate	BTL-5
3.	**Design** a consecutive integer checking algorithm and middle-school procedure algorithm. (15)	Create	BTL-6
4.	**Design** the steps of mathematical analysis for recursive and non-recursive algorithm with an example of each. (15)	Create	BTL-6
	Give the backward substituion plan to find the Time Efficiency of Recursive Algorithms and use recurrence to find number of moves for Towers of Hanoi problem. (15)	Create	BTL-6

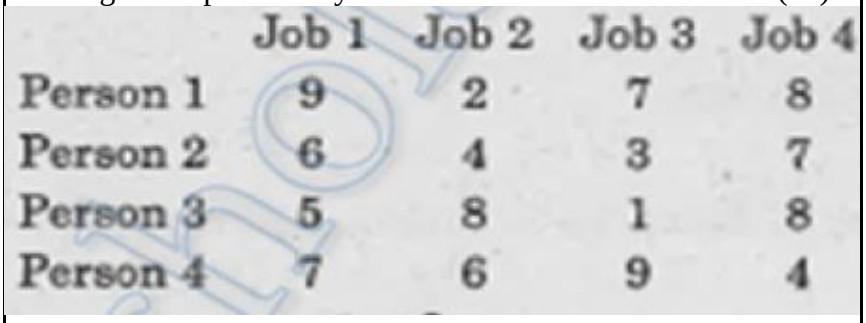
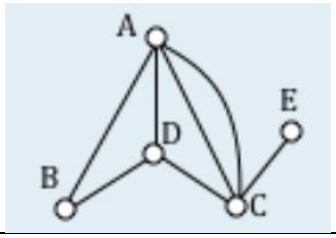
UNIT II - BRUTE FORCE AND DIVIDE-AND-CONQUE			
Brute Force – Computing an – String Matching - Closest-Pair and Convex-Hull Problems - Exhaustive Search - Travelling Salesman Problem - Knapsack Problem - Assignment problem.			
PART – A			
Q.No	**Questions**	**BT Level**	**Competence**
1.	**State** Brute force approach.	Remember	BTL-1
2.	**Examine** a brute force algorithm for string matching problem.	Apply	BTL-3
3.	Give an example of a text of length n and a pattern of length m that constitutes a worst case input for the brute force string matching algorithm. **Formulate** and find how many character comparisons will be made for such input.	Create	BTL-6
4.	**Define** the term exhaustive search.	Remember	BTL-1
5.	**Examine** a brute force algorithm for counting the number of vowels in a given text.	Apply	BTL-3
6.	**Give** an example problem that cannot be solved by a Brute force approach and also how to decide?	Remember	BTL-1
7.	**Describe** brute force approach. What are the advantages and disadvantages of this approach?	Analyze	BTL-4

8.	**Define** closest pair problem.	Remember	BTL-1
9.	**Define** convex hull problem.	Remember	BTL-1
10.	**Formulate** is the length of the step-in jump search?	Create	BTL-6
11.	**Analyze** the objective of the knapsack problem?	Analyze	BTL-4
12.	**Describe** the concepts of Travelling Salesman Problem.	Analyze	BTL-4
13.	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? of the following methods can be used to solve the Knapsack problem?	Understand	BTL-2
14.	What is the **time complexity** of the brute force algorithm used to solve the Knapsack problem?	Apply	BTL-3
15.	**Define** the Time complexity of knapsack 0/1 where n is the number of items and W is the capacity of knapsack.	Understand	BTL-2
16.	**Give** an example for knapsack problem.	Understand	BTL-2
17.	**Evaluate** the methods can be used to solve the Knapsack problem?	Evaluate	BTL-5
18.	Given items as {value,weight} pairs {{60,20},{50,25},{20,5}}. The capacity of knapsack=40. **Find** the maximum value output assuming items to be divisible and non-divisible respectively.	Evaluate	BTL-5
19.	**Define** Assignment problem.	Remember	BTL-1
20.	**What** is the advantage of recursive approach than an iterative approach?	Understand	BTL-2
21.	**Give** an example for a Assignment problem.	Understand	BTL-2
22.	**Analyze** the concpt of shorest hamiltonian circuit problem?	Apply	BTL-3
23.	**Give** the basic operation involved in finding the closest pair problem.	Understand	BTL-2
24.	**Formulate** Euclidean distance.		
	PART – B		
1.	**Explain the concepts** of the following. (i)Brute force string matching Algorithm. (7) (ii)Closest pair and convex hull problems by brute force (6)	Evaluate	BTL-5
2.	Can you design a more efficient algorithm than the one based on the brute-force strategy to solve the closest-pair problem for n points x1, x2,...,xn on the real line? (13)	Understand	BTL-2
3.	i)What is the closet pair problem? (3) ii)Explain the brute force approach to solve closest-pair with an example. (6) iii)Derive the time complexity.	Analyze	BTL-4
4.	What is brute force strategy and explain the sequential search with suitable example problem. (13)	Analyze	BTL-4

5.	**Examine** in detail about Exhaustive search techniques. (13)	Remember	BTL -1
6.	i)What is the convex hull problem? (3) ii)Explain the brute force approach to solve convex-hull with an example. (6) iii)Derive the time complexity. (4)	Remember	BTL-1
7.	Determine the number of character comparisons made by the brute-force algorithm in searching for the pattern GANDHI in the text THERE_IS_MORE_TO_LIFE_THAN_INCREASING_ITS_SPEED Assume that the length of the text—it is 47 characters long—is known before the search starts (13)	Create	BTL-6
8.	How many comparisons (both successful and unsuccessful) will be made by the brute-force algorithm in searching for each of the following patterns in the binary text of one thousand zeros? (13)	Remember	BTL-1
9.	Explain the Bruteforce method to find the two closest points in a set of n points in k-dimensional space. (13)	Apply	BTL-3
10.	Solve travelling sales man problem using brute force technique. (13)  	Apply	BTL-3
11.	Solve shortest hamiltonian circuit problem using brute force technique. (13)  	Apply	BTL-3
12.	Explain the brute force method for the travelling sales man problem and give the algorithm with your own example.	Understand	BTL-2
13.	Explain the brute force method for the knapsack problem and give the algorithm. (13)	Understand	BTL-2
14.	(i)List out the procedures to solve travelling salesman problem. (6) (ii)Describe the Knapsack problem by using Exhaustive search. (7)	Understand	BTL-2
15.	Find and Analyze the optimal solution for the assignment problem given below. (13) Job 1 Job 2 Job 3 Job 4 4 3 8 6 5 7 2 4 16 9 3 1 2 5 3 7	Analyze	BTL-4

16.	Define Assignment problem. Examine the optimal solution for the assignment problem with one example (13)	Remember	BTL-1
17.	Give an example of the assignment problem whose optimal solution does not include the smallest element of its cost matrix. (13)	Evaluate	BTL-5

PART – C

1.	Solve using Brute force approach to **evaluate** and find whether the given string follows the specified pattern and return 0 or 1 accordingly. Examples: 1)Pattern "abba" input: "redblueredblue" should return 1 2)Pattern "aaaa" input: "asdasdasdasd" should return 1 3)Pattern "aabb" input: "xyzabcxyzabc" " should return 0 (15)	Evaluate	BTL-5
2.	Consider the problem of counting, in a given text, the number of substrings that start with an A and end with a B. For example, there are four such substrings in CABAAXBYA. i). Design a brute-force algorithm for this problem and determine its efficiency class. (8) ii) Design a more efficient algorithm (7)	Evaluate	BTL-5
3.	There are 4 people who need to be assigned to execute 4 jobs(one person per job) and the problem to find an assignment with the minimum total cost. The assignment costs is given below, solve the assignment problem by exhaustive search. (15)  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	Create	BTL-6
4.	**Formulate** and give an example of a text of length *n* and a pattern of length *m* that constitutes a worst-case input for the brute-force string-matching algorithm. Exactly how many character comparisons will be made for such input? (15)	Create	BTL-6
5.	Does a hamiltonian path or circuit exists on the graph below problem using brute force technique. (15)  	Create	BTL-6

UNIT III - DIVIDE-AND-CONQUER TECHNIQUE			
Divide and Conquer Methodology – Binary Search – Merge sort – Quick sort – Heap Sort - Multiplication of Large Integers– Closest-Pair and Convex - Hull Problems.			
PART – A			
Q. No	**Questions**	**BT Level**	**Competence**
1.	**Define** Divide and conquer methodology.	Remember	BTL-1
2.	**List the** sorting methods would be most suitable for sorting a list.	Remember	BTL-1
3.	**Prove** asymptotic complexity in terms of n.	Apply	BTL-3
4.	**Evaluate** total number of comparisons made in quick sort for sorting a file of size n.	Evaluate	BTL-5
5.	**Design** the correct order of the efficiency of the following sorting algorithms according to their overall running time comparison.	Create	BTL-6
6.	**How** do you measure the efficiency of an algorithm for divide and conquer method?	Remember	BTL-1
7.	**Examine** an algorithm for binary search.	Apply	BTL-3
8.	**Identify** The number of swapping's needed to sort the numbers 8, 22, 7, 9, 31, 19, 5, 13 in ascending order, using bubble sort.	Remember	BTL-1
9.	How many approaches can be applied to solve quick hull problem	Analyze	BTL-4
10.	**Explain** the various types of problems does quick hull belong.	Analyze	BTL-4
11.	**Apply** the common technique for proving to find the 'n' points that lie in a convex quadrilateral.	Apply	BTL-3
12.	**Define** space complexity of merge sort.	Remember	BTL-1
13.	**Define** worst case time complexity of merge sort.	Remember	BTL-1
14.	**Formulate** the Pattern of Problems in Divide and Conquer approach	Create	BTL-6
15.	**Differentiate** between Merge sort and Heap sort efficiency.	Understand	BTL-2
16.	**Discuss** the concepts of Quick sort and its properties.	Understand	BTL-2
17.	**What** is heap sort explain with example?	Apply	BTL-3
18.	**Examine** an recurrence relation for multiplication of large numbers.	Evaluate	BTL-5
19.	**What** is the first step of heap sort?	Understand	BTL-2
20.	**Analyze** quick sort with example?	Analyze	BTL-4
21.	**Analyze** the Run Time of Merge Sort .	Analyze	BTL-4
22.	**Evaluate** the Time Complexity of binary search tree.	Evaluate	BTL-5
23.	A machine needs a minimum of 200 sec to sort 1000 elements by Quick sort. **What** is the minimum time needed to sort 200 elements will be approximately	Understand	BTL-2
24.	Apply Quick sort on a given sequence 7 11 14 6 9 4 3 12. What is the sequence after first phase, pivot is first element?	Understand	BTL-2

	PART-B			
1	**Explain** in detail about the closest pair and convex hull problems by using Divide and conquer method.	13	BTL -1	Remember
2	**Analyze** in detail about divide and conquer strategy with a scenario.	13	BTL -3	Apply
3	What is divide and conquer strategy and **explain** the binary search with suitable example problem.	13	BTL -4	Analyze
4	What is divide and conquer strategy and **explain** the multiplication of large numbers with suitable example problem.	13	BTL -4	Analyze
5	**Illustrate** an algorithm for Quicksort and write its time complexity with example list are 5,3,1,9,8,2,4,7.	13	BTL -3	Apply
6	**Describe** in detail about the operation of binary search algorithm for the input -15, -6 , 0, 7 , 9, 23, 54, 82, 101,112, 125,131,142,151 if you are searching for the element 9.	13	BTL -2	Understand
7	**Analyze** the pros and cons of convex hull problem and the solution involved in detail.	13	BTL -4	Analyze
8	**Discuss** in detail and write an algorithm to sort a given list of elements using heap sort. Show the operation of the algorithm, on the list 14,12,9,8,7,10,8.	13	BTL -2	Understand
9	Write the algorithm for quicksort. Provide a complete analysis of quick sort for the given set of numbers 12,33,23,43,44,55,64,77 and 76.	13	BTL -5	Evaluate
10	**Describe** in details about the three processing steps in Quick sort.	13	BTL -1	Remember
11	**Write** the algorithm to perform the working of multiplication of large numbers with an example.	13	BTL -1	Remember
12	(i)Find the number of comparisons required to search for '6' in the given Sequence of numbers: 10, 19, 7, 9, 6, 15. (ii)Analyze the time efficiency and drawbacks of merge sort algorithm.	7 6	BTL -3	Apply
13	Write a program implementing the divide-conquer algorithm for the closest pair problem.	13	BTL -1	Remember
14	Write the mergesort algorithm and explain it with an example. Derive the worst case and average case time complexity.	13	BTL -2	Understand
15	(i) Multiply 23 * 14 using divide and conquer strategy (ii)Compute the complexity for multiplication of large numbers.	7 6	BTL -6	Create
16	Write the max-heap algorithm and explain it with an example. Derive the worst case and average case time complexity.	13	BTL -6	Create
17	Write a program implementing the divide-conquer algorithm for the convex-hull problem.	13	BTL -2	Understand

	PART – C		
1.	Solve heap-sort by applying the Divide and Conquer method and Analyze the time and space complexity of Divide and conquer methodology for the array having elements: 16, 10, 15, 9, 5, 12, 14.	Create	BTL-6
2.	Solve 2138 × 4967 by applying the Divide and Conquer method and Analyze the time and space complexity of Divide and conquer methodology	Evaluate	BTL-5

3.	Examine that the procedure SEARCH of binary search algorithm gives the Smallest possible expected search time if all elements in the universal set are equally likely to be sought. (15)	Evaluate	BTL-5
4.	(i) Design a Quick sort algorithm (7) (ii) Develop Best, worst and Average case analysis for Quicksort method. (8)	Create	BTL-6
5.	Discuss the topic on merge sort. Illustrate the algorithm with numeric Example. Predict the complete analysis for the same. Is merge sort stable sorting algorithm? Justify your answer. (15)	Create	BTL-6

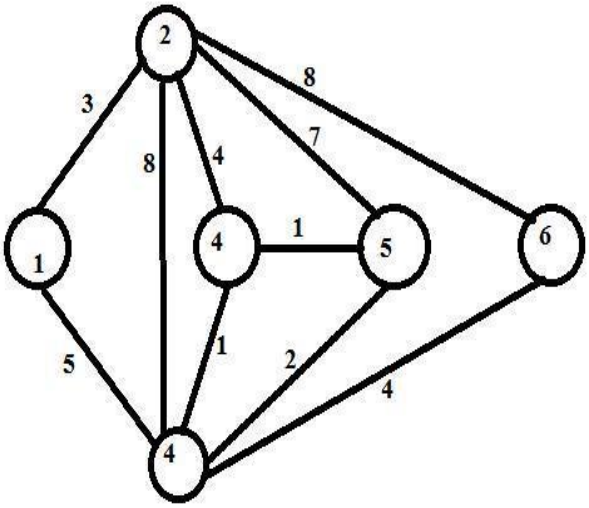
UNIT IV - DYNAMIC PROGRAMMING AND GREEDY TECHNIQUE

Dynamic programming – Principle of optimality - Coin changing problem, Computing a Binomial Coefficient – Floyd‘s algorithm – Multi stage graph - Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique – Container loading problem - Prim‘s algorithm and Kruskal's Algorithm – 0/1 Knapsack problem, Optimal Merge pattern - Huffman Trees.

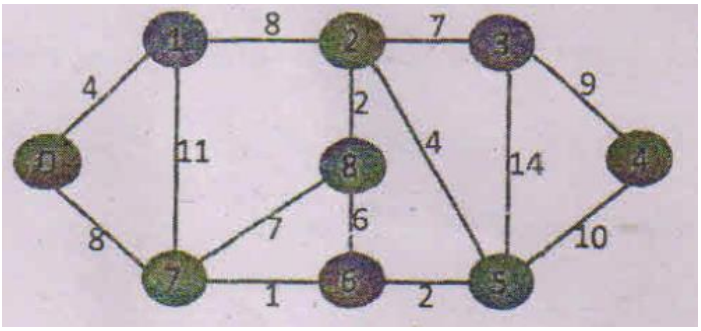
PART – A

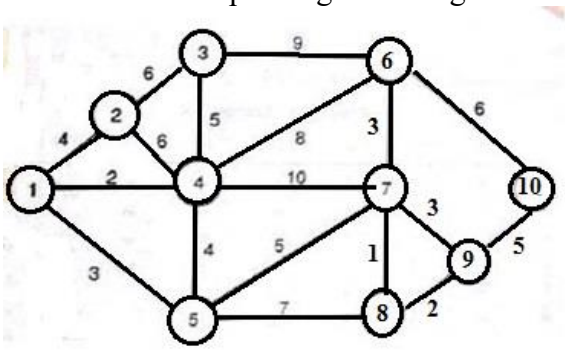
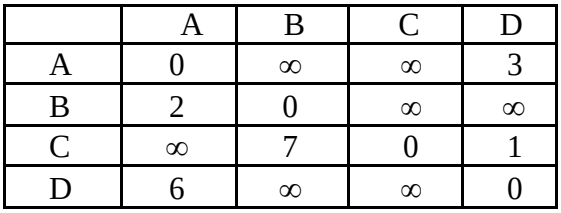
Q. No	**Questions**	**BT Level**	**Competence**
1.	**What** is dynamic programming?	Remember	BTL-1
2.	**How** is a transportation network represented?	Apply	BTL-3
3.	**Compare** memorization method vs tabulation method.	Analyze	BTL-4
4.	**Estimate** the space and time complexity of computing a Binomial Coefficient.	Understand	BTL-2
5.	**State** the principle of optimality.	Remember	BTL-1
6.	**Define** Transitive closure of a directed graph.	Remember	BTL-1
7.	**What** is the constraint for binary search tree insertion?	Remember	BTL-1
8.	**Compare** Divide & Conquer and Dynamic Programming.	Apply	BTL-3
9.	**Discover** the pseudo code of the Warshall’s algorithm.	Apply	BTL-3
10.	**Summarize** feasible and optimal solution.	Understand	BTL-2
11.	**Contrast** Greedy algorithm and Dynamic programming.	Apply	BTL-3
12.	**List** the properties of Dynamic programming approach	Remember	BTL-1
13.	**Define** the minimum spanning tree problem.	Remember	BTL-1
14.	**Explain** how the Binomial coefficient is computed.	Evaluate	BTL-5
15.	**Estimate** the time and space complexity for Warshall’s algorithm.	Understand	BTL-2
16.	**Demonstrate** the obstacles in constructing a minimum spanning	Apply	BTL-3

	tree by an exhaustive search.		
17.	**Estimate** the space and time complexity of a prim's algorithm.	Understand	BTL-2
18.	**Analyze** the time complexity of optimal Binary search Tree algorithm.	Analyze	BTL-4
19.	**Distinguish** prim's and Kruskal's algorithm.	Understand	BTL-2
20.	**Summarize** Huffman trees and its applications	Evaluate	BTL-5
21.	**Integrate** Minimum spanning tree concepts and Prim's algorithm.	Create	BTL-6
22.	**Develop** an algorithm for memory function knapsack problem.	Create	BTL-6
23.	**Analyze** the time complexity for Binomial coefficient algorithm.	Analyze	BTL-4
24.	**Analyze** the time complexity of multi stage graph algorithm	Analyze	BTL-4

1.	Consider the following distance network. a) Write the floyd's algorithm and generate the final distance matrix. (7) b) **Analyze** the shortest path and the corresponding distance from the source node to the destination node as indicated in each of the cases 1-6, 5-1 and 5-2 (6)  	Analyze	BTL-4
2.	**i) Illustrate** all-pair shortest path problem algorithm. (4) **(ii)Calculate** the all-pair shortest path problem for the diagraph with the weighted matrix given below. (9)	Apply	BTL-3

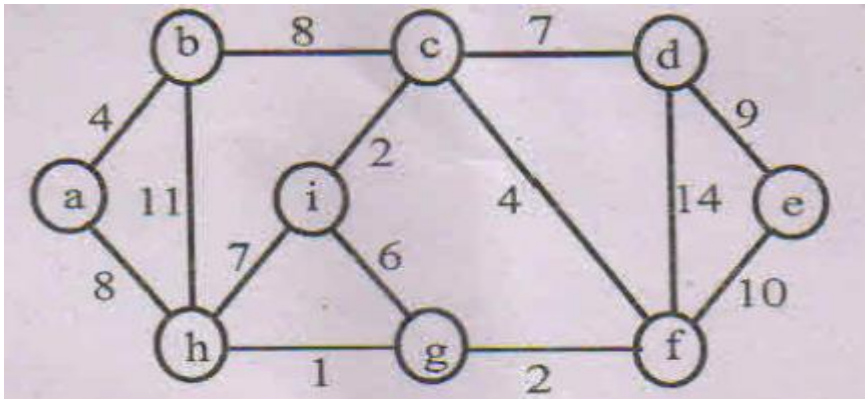
	a	b	c	d
a	0	α	3	α
b	2	0	α	α
c	A	7	0	1
d	6	α	α	0

3.	**Explain** how multistage graph for finding the single-source shortest paths for the given graph. (13) 	Understand	BTL-2
4.	**Describe** and compute binomial coefficient by the formula C(n, k) = C(n − 1, k − 1) +C(n − 1, k). (13)	Analyze	BTL-4
5.	**Analyze** the algorithm by applying the following keys and probabilities to obtain the optimal binary tree. (13) Key: A, B, C, D Probability: 0.1, 0.2, 0.4, 0.3	Evaluate	BTL-5
6.	i) Consider 4 elements a1< a2< a3< a4with q0=0.25, q1=3/16, q2=q3=q4=1/16, P1=1/4, P2=1/8, P3=P4=1/16. Construct the optimal binary search tree as a minimum cost tree (7) ii) Construct the table of values Wij, Cij, Vij computed by the algorithm to compute the roots of optimal sub trees. (6)	Evaluate	BTL-5
7.	Plan the following instance of the 0/1, knapsack problem given the knapsack capacity in W=5 using dynamic programming and explain it. Item 1: Weight 4, Value $10 Item 2: Weight 3, Value $20 Item 3: Weight 2, Value $15 Item 4: Weight 5, Value $25	Create	BTL-6
8.	(i)Define Huffman tree. List the types of Encoding in Huffman tree. (8) (ii)Write the Huffman's algorithm. Construct the Huffman's tree for the following data and obtain its Huffman code. (5) Character: A, B, C, D, E, _(Underscore) Probability: 0.5, 0.35, 0.5, 0.1, 0.4, 0.2	Remember	BTL-1
9.	(i) Describe minimum spanning tree using Kruskal's algorithm with an example. (7) (ii) Comparison between Prim's and Kruskal's algorithm and identify the time complexity of those algorithms. (6)	Remember	BTL-1
10.	**(i)**Write and analyze the prim's algorithm. (5)	Remember	BTL-1

	(ii)Describe minimum spanning tree using Prim's algorithm. (8) 		
11.	**(i)List** out the short notes on optimal binary search tree. (7) **(ii) Label** the optimization technique used for optimal binary search algorithm. (6)	Remember	BTL-1
12.	Explain the steps in building Huffman Tree.Find the codes for the alphabets given below according to the frequency. Let _(Space)= 4 A= 2 , E = 5 , H = 1, I = 2, L = 2, M = 2 , P = 2 R = 1,S =2 , X = 1 (13)	Analyze	BTL-4
13.	**(i) Examine** Coin changing problem with a suitable example (7) **(ii)Illustrate** how the problem can be solved by dynamic programming approach. (6)	Apply	BTL-3
14.	**Compare** memorization method vs tabulation method with an example.	Apply	BTL-3
15.	**Explain** 0/1 knapsack problem using dynamic programming approach in detail.	Understand	BTL-2
16.	**Explain** multistage graph and explain memory functions problem in detail. (13)	Apply	BTL-3
17.	**Illustrate** how the problem of optimal merge pattern can be solved using dynamic programming approach.	Apply	BTL-3
	PART – C		
1.	**Asses** and solve all-pair shortest path problem for the digraph with the weight matrix given below: (15) 	Evaluate	BTL-5

2.	(i)What is optimal merge pattern? (4) (ii)How greedy method is useful to obtain an optimal merge pattern. (6) (iii)Give the example of optimal merge pattern. (5)	Evaluate	BTL-5
3.	Apply Warshall's algorithm to find the transitive closure of the	Create	BTL-6

	A	B	C	D
A	0	∞	∞	3
B	2	0	∞	∞
C	∞	7	0	1
D	6	∞	∞	0

	digraph defined by the following adjacency matrix $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ i) Prove that the time efficiency of Warshall's algorithm is cubic (7) ii) **Explain** why the time efficiency of Warshall's algorithm is inferior to that of the traversal-based algorithm for sparse graphs represented by their adjacency lists. (8)		
4.	Apply the greedy technique to find the minimum spanning tree using Prim's algorithm for the given graph. (15) 	Create	BTL-6
5.	Apply the technique to reduce the size of the given message using huffman coding technique. (Fized size & Variable Size) Message : BCCABBDDAECCBBAEDDCC	Create	BTL-6

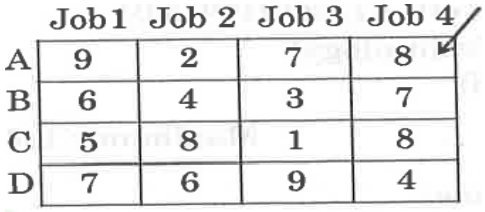
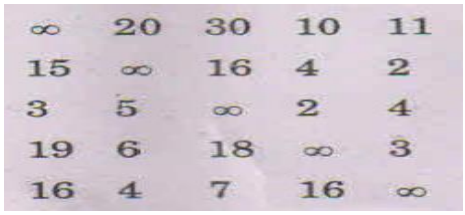
20.	**Give** some examples of P and NP problem.	Understand	BTL-2

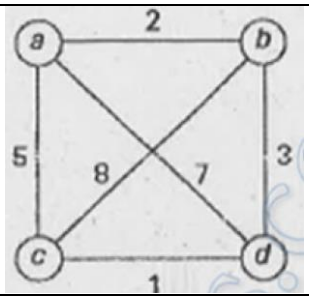
UNIT V - COPING WITH THE LIMITATIONS OF ALGORITHM POWER

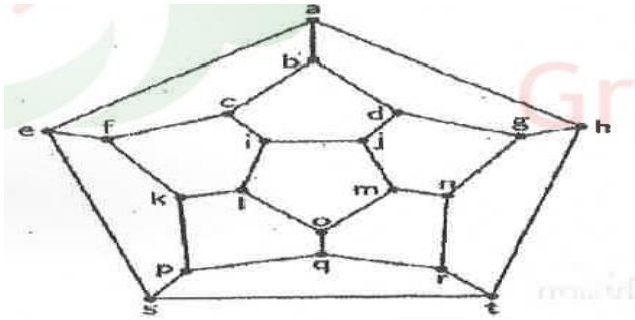
Lower - Bound Arguments - P, NP NP- Complete and NP Hard Problems. Backtracking – n-Queen problem - Hamiltonian Circuit Problem – Subset Sum Problem. Branch and Bound – LIFO Search and FIFO search - Assignment problem – Knapsack Problem – Travelling Salesman Problem - Approximation Algorithms for NP-Hard Problems – Travelling Salesman problem – Knapsack problem.

PART – A

Q. No	Questions	BT Level	Competence
1.	**What** are tractable and non-tractable problems?	Remember	BTL-1
2.	**Compare** class P and class NP.	Analyze	BTL-4
3.	**Define** NP hard problem.	Remember	BTL-1
4.	**Discuss** the principle of backtracking.	Understand	BTL-2
5.	How is the accuracy of approximation algorithm **measured**?	Evaluate	BTL-5
6.	**State** Hamiltonian Circuit problem.	Remember	BTL-1
7.	What are the additional items required for branch and bound? **compare** backtracking technique.	Analyze	BTL-4
8.	**Point out** some examples of lower bound.	Analyze	BTL-4
9.	**Describe** the term state space tree.	Remember	BTL-1
10.	**Define** Knapsack problem.	Remember	BTL-1
11.	**Discuss** the term best first branch bound.	Understand	BTL-2
12.	**State** whether backtracking always produces optimal.	Create	BTL-6
13.	**Decide** the termination point of the search path in a state space tree of branch and bound algorithm.	Evaluate	BTL-5
14.	**Show** formal definition of the n-queens problem.	Understand	BTL-2
15.	**Describe** the term state space tree.	Understand	BTL-2
16.	What is Hamiltonian path? **Generalize** that Hamiltonian cycle is an undirected graph.	Create	BTL-6
17.	What does NP-hard mean? **Demonstrate** approximation algorithm for NP hard problem.	Apply	BTL-3
18.	**How** is lower bound found by problem reduction?	Understand	BTL-2
19.	**Examine** the subset sum problem.	Analyze	BTL-4
20.	**Define** NP complete problem.	Evaluate	BTL-5
21.	**Compare** polynomial time vs Exponential time.	Apply	BTL-3
22.	**How** maximum matching is done in a Bipartie graph**?**	Remember	BTL-1
23.	**Describe** stable marraige problem.	Apply	BTL-3
24.	**State** 8 Queens problem.	Apply	BTL-3

	PART-B			
1.	**Explain** in detail about the Backtracking with an example.	13	Remember	BTL-1
2.	(i) **Evaluate** the subset sum problem with set as {3, 5, 6, 7, 2}and the sum =15.Derive all the subsets. (ii)	7 6	Evaluate	BTL-5
3.	(i) **Identify** an example for the best-case input for the branch and bound algorithm for the assignment problem. (ii) Describe NP-hard and NP-completeness.	7 6	Apply	BTL-3
4.	Using Back-Tracking enumerate how can you **solve** the following problems. (i)4-queens problem. (ii)Hamiltonian circuit problem.	7 6	Apply	BTL-3
5.	Find the optimal solution using Branch and Bound for the following assignment problem.  	13	Apply	BTL-3
6.	Consider the travelling salesperson instance defined by the following cost matrix.  Draw the state space tree and show the reduced matrices corresponding to each of the node.	13	Understand	BTL-2
7.	(i)**Explain** how to implement an algorithm for Knapsack problem using NP-Hard approach. (ii)**Distinguish** between the P and NP problems.	7 6	Analyze	BTL-4
8.	**Describe** about the following: (i)Approximation algorithms for the knapsack problem. (ii) Approximation algorithms for Traveling sales man problem.	6 7	Remember	BTL-1
9.	(i)Explain the backtracking problem. (ii) Analyze and explain elaborately on backtracking algorithm.	7 6	Analyze	BTL-4
10.	**Write** an algorithm for subset sum and explain with an example.	13	Apply	BTL-2
11.	**Explain** in detail about assignment problem using branch and bound technique.	13	Remember	BTL-1
12.	**Apply** Branch and Bound algorithm to solve the travelling salesman problem for	13	Create	BTL-6

				
13.	With an example, **summarize** how the branch and bound technique is used to solve 0/1 knapsack problem.	13	Evaluate	BTL-5
14.	There are 5 distinct numbers {1,2,5,6,8}. **Identify** the combinations of these numbers such that the sum is 9. Use the backtracking model to arrive at the solution.	13	Understand	BTL-2
15.	With an example, **Exaplin** how the backtracking approach is used to solve Hamiltonian circuit problem.	13	Analyze	BTL-4
16.	**Explain** in detail about the LIFO and FIFO search.	13	Remember	BTL-1
17.	**Write** down the difference between Backtracking and Brach and Bound**.**	13	Understand	BTL-2

	PART-C			
1.	**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 constraints.	15	Create	BTL-6
2.	**Find a** Hamiltonian circuit or disprove its existence in the graph given below. 	15	Evaluate	BTL-5
3.	**Solve** the following instance of Knapsack problem by branch and bound algorithm. Item weight profit 1 5 $40 2 7 $35 3 2 $18 W=15 4 4 $4 5 5 $10 6 1 $2	15	Create	BTL-6
4.	Let w={5,7,10,12,15,18,20} and m=35.**Compute** all possible subset of w whose sum is equivalent to m. Draw the portion of state space tree for this problem.	15	Evaluate	BTL-5
5.	Evaluate the following instance of the knapsack problem using the branch and bound algorithm. Knapsack capacity W=10. Item Weight Value 1 4 $40 2 7 $42 3 5 $25 4 3 $12	15	Evaluate	BTL-5