This sheet is for 1 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	What is randomized algorithms?		Used to solve many time consuming problems	Used to solve shotesh path algorithms	Both A and B	Used to solve tactable problems	a
2	What are the loop invarient propoerties?		1) Time complexity 2) Space complexity 3) Memory	1) RAM 2) ROM 3) Hard Disk	Both A and B	1) Initialization 2) Maintenance 3) Termination	d
3	Why correctness of algorithm is essential?		To check for efficiency	To check the execution time	To verify the space required	To check whether output produced by algorithm is correct	d
4	Examples of O(n2) algorithms are___.	img.jpg	Adding of two Matrices	Initializing all elements of matrix by zero	Both A and B	Neither A nor B	c
5	Two main measures for the efficiency of an algorithm are	img.jpg	Processor and memory	Complexity and capacity	Time and space	Data and space	c
6	What are the main features that for algorithm works as technology?		Time Complexity	Space Complexity	Both A and B	no of the above	c
7	What is complexity theory?		study of algorithm is known as complexity theory	study of time and space complexity know as complexity theory	Study of tactable and non tactable problems is known as complextiy theory	no the above	c
8	What is the correctness of algorithm?		Error free code	Every instance will be producing correct result	No of lines is minimum	Time taken by steps is absolute	b
9	When determining the efficiency of algorithm the time factor is measured by	img.jpg	Counting microseconds	Counting the number of key operations	Counting the number of statements	Counting the kilobytes of algorithm	b
10	Examples of O(1) algorithms are___	img.jpg	Multiplying two numbers.	Assigning some value to a variable	Displaying some integer on console	All of the above	d
11	What are the main features that for algorithm works as technology?		Time Complexity	Space Complexity	Both A and B	no of the above	c
12	Effectivity of writing algorithms includes		1) Removing reduendant operations 2) Referencing array 3) Late termination	1) Removing errors 2) Reducing lines of code.	1) Removing functions. 2) Removing errors	None of the above	a
13	Iterative algorithm design issues are ?		1) Initialization 2) Maintenance 3) Termination	1) Time complexity 2) Space complexity 3) Memory	1) Initail condition 2) Iterative construct 3) Loop termination	None of the above	c
14	When an operation typically takes no more than a fixed amount of time, we can say that its time is bounded by a		Constant	variable	function	cant masure	a
15	In a rcursion tree , each node reprsents a cost of a		Multiple sub problem	Single sub problem	whole problem	non of the abov	b
10	In Huffman coding, data in a tree always occur?		roots	leaves	left sub tree	right sub tree	b

2 Marks

S.r No	Question	Image	a	b	c	d	Correct Answer
1	Which of the following is not a property of an Algorithm?		Finitenss	Deifiniteness	Effectiveness	Incompleteness	d
2	The main measure for efficiency of an algorithm are:		Data and space	Complexity and capacity	Input and output	Time and space	d
3	O(1) means that the number of executions of basic operations is		Variable	Fixed	Not defined	polynomial	b
4	In the iteration method we iteratively _____ the recurrence until we see the pattern		substract	add	unfold	pack	c
5	Using the substitution method, it is easy to prove a		Exact bound	strong bound	weaker bound	none of the above	d
6	What is a principal of optimality?		An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision	Optimal solution from set	Generat optmal solution from the set of efficient optimum sets	All of the above	a
7	Following is true for understanding of a problem		Knowing the knowledgebase	Understanding the subject on which the problem is based	Communication with the client	All of the above	d
8	Which of the above is true		O(n) < O(log n)< O(n logn) < O(n) <O(n^2)<O(n^3)< O(2^n)	O(n) < O(n)< O(logn) < O(n log n) <O(n^2)<O(n^3)< O(2^n)	O(n) < O(log n)< O(n) < O(n log n) <O(n^2)<O(n^3)< O(2^n)	O(n) < O(log n)< O(n) < O(n log n) <O(n^2)<O(2^n)<O(n^3)	c
9	What is the average case time complexity of binary search using recursion?		O(Log n)	O(n Log n)	O(n)	O(n2)	a
10	In Huffman coding, data in a tree always occur?		roots	leaves	left sub tree	right sub tree	b

3 marks

3 Marks MCQ Question

S.r No	Question	Image	a	b	c	d	Correct Answer

1	How many times is the comparison i >= n performed in the following program? int i = 200, n = 80;
	main(){
	while (i >= n){
	i = i-2;
	n = n+1;
	}
	}
			20	30	42	50	c

2	Which of the following time complexities best describes MaxMin approach		Θ(n)	Ω(n)	O(nlogn)	O(logn)	a
3	Which one of the following is the recurrence equation for the worst case time complexity of theQuicksort algorithm for sorting n(≥2) numbers? In the recurrence equations given in the optionsbelow, c is a constant.		T(n) = 2T(n/2) + cn	T(n) = T(n−1) + T(1) + cn	T(n) = 2T(n−2) + cn	T(n) = T(n/2) + cn	b

4	Write the pseudo code for finding the factorial of given number.		int Recursive_fact(int n) {
			return 1;
			}
			else
			{
			return (n*Recursive_fact(n-1));
			}
			
			If (n==0) then
			{
				int Recursive_fact(int n)
				{
				return 1;
				}
				else
				{
				return (n+Recursive_fact(n-1));
				}
				
				If (n==0) then
				{
					int Recursive_fact(int n)
					{
					return 1;
					}
					else
					{
					return (Recursive_fact(n-1));
					}
					
					If (n==0) then
					{
						none of the above is correct	a

5	Consider a set of 4 messages (M1 – M4) whose frequency of occurrences in the text is as given: (0.37, 0.51, 0.05, 0.07) Using frequency dependent Huffman Coding the codes of the messages M2 and M3 respectively.		0, 110	0, 011	1, 000	1, 001	c

This sheet is for 1 Mark questions

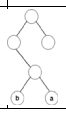
S.r No	Question	Image	a	b	c	d	Correct Answer

1	Solve the following recurrence using Master's theorem. T(n) = 4T (n/2) + n2		T(n) = O(n)	T(n) = O(log n)	T(n) = O(n2log n)	T(n) = O(n2)	c -The given recurrence can be solved by using the second case of Master's theorem. T(n) = O(nc log n) = Here nc = n2 So the solution becomes T(n) = O(n2log n).
2	What will be the recurrence relation of the following code? Int sum(int n) { If(n==1) return 1; else return n+sum(n-1); }		T(n) = T(n/2) + n	T(n) = T(n-1) + n	T(n) = T(n-1) + O(1)	T(n) = T(n/2) + O(1)	c Explanation -The given recurrence can be solved by using the second case of Master's theorem. T(n) = O(nc log n) = Here nc = n2 So the solution becomes T(n) = O(n2log n).
3	Freactinal knpsack problem is		0/1 knapsack problem	continuous knapsack probl	Divisible knapsack proble	Non continuoue knapsack pro	b
4	The Huffman code length does not depend on the frequency of occurrence of characters		TRUE	FALSE	cant say	variabl length	b
5	What is tail recursion?		Recursiv explicite function	Dynamic function of execution	First function at execution know as tail recursion	A recursive function is tail recursive when recursive call is the last thing executed by the function	d
6	The time complexity of linear search is_____.		O(1)	O(log n)	O(n)	O(n logn)	c
7	The concept of order Big O is important because		It can be used to decide the best algorithm that solves a given problem	It determines the maximum size of a problem that can be solved in a given amount of time	It is the lower bound of the growth rate of algorithm	All of the above	a
8	What is greedy approach for solving problems?		A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum.	An greedy algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a requiredoutput for any legitimate input in finite amount of time.An algorithm is step by step procedure to solve a problem	A straightforward approach, usually based directly on the problem s statement and definitions of the concepts involved.	None of the above	a
9	The integer value i in the list of values name[i] is also known as?		Marker	index	condition	None of the above	b
10	The Big O notation defines an _____ bound of an algorithm		upper	lower	middle	None of the above	a
11	How many cases are there in masters theoram		2	3	4	5	b
12	We can solve any type of recurrence relation		TRUE	FALSE	cant say	partially	b
13	Features of functional model		1) Primitive expression 2) Defination of one function in terms of other 3) Defination of functions using conditions 4) Inductive defination of functions	1) Substitution of functions 2) Inductive defination of function	1) Initial condition model 2) Iterative construct model 3) Loop model	It does not represented using model	a
14	Identify which statement is correct related to recursion		Recursion is procedure oriented execution	Recursion is computational process and charactrise by chain of deferred oprations	Recursion is function to funtion call	Recursion is explicite call to functions	b
15	The time complexity of binary search is_____.		O(1)	O(log n)	O(n)	O(n logn)	b

This sheet is for 2 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	Fractional knapsack problem is solved most efficently by which algorithm		Divide and Conqure	Dyanamic Programming	Greedy Algorithm	Backtracking	c
2	Fractional Knapsack approach computes		Weight per unit profit	Profit per unit weight	Product of profit and w	None of these	b
3	Usually a recurrence reflects the runtime of a		Non-recursive algorithm	Infinite algorithm	Recursive algorithm	None of the above	c
4	Following is true for understanding of a problem		Knowing the knowledgebase	Understanding the subject on which the problem is based	Communication with the client	All of the above	d
5	Huffman code generation can be effectively solved by which mehtod		Dynamin Programming	Branch and Bound	Greedy Algorithm	cant say	c
6	Locally declared variable can be accessed outside the function		TRUE	FALSE	cant say	no of the above	a
7	Globally declared variables can be accessed through out the program		FALSE	TRUE	partially	cant say	b
8	An algorithm which is exponential will be practical only for _____ values of n		large	very large	small	very small	b
9	The different cases of Master's theorem to solve recurrences are		exhaustive	non exhaustive	can not dfine	All of the above	a
10	Master theoraem is used for		solving recurrence	solving iterative relations	analysing loops	calculating time complexity of code	a
6	Usually a recurrence reflects the runtime of a		Non-recursive algorithm	Infinite algorithm	Recursive algorithm	None of the above	c
7	Fractional Knapsack approach computes		Weight per unit profit	Profit per unit weight	Product of profit and w	None of these	b
8	Huffman code generation can be effectively solved by which mehtod		Dynamin Programming	Branch and Bound	Greedy Algorithm	cant say	c
9	Locally declared variable can be accessed outside the function		TRUE	FALSE	cant say	no of the above	a
10	Globally declared variables can be accessed through out the program		FALSE	TRUE	partially	cant say	b

3 Marks MCQ Question

S.r No	Question	Image	a	b	c	d	Correct Answer
1	The complexity of fractional knapsack problem		O(n log n)	O(n)	O(n2)	O(nW)	a
2	Which of the following algorithms is the best approach for solving Huffman codes?		exhaustive search	greedy algorithm	brute force algorithm	divide and conquer algorithm	b
3	If two jobs J1 and J2 have same deadline but profit of J2 is more than profit of J1 which among the two jobs will achieve maximum deadline		J1	J2	either J1 or J2	Does not depend on profit and deadline	J2
4	Which one of the following is the recurrence equation for the worst case time complexity of theQuicksort algorithm for sorting n(≥2) numbers? In the recurrence equations given in the optionsbelow, c is a constant.	img.jpg	T(n) = 2T(n/2) + cn	T(n) = T(n−1) + T(1) + cn	T(n) = 2T(n−2) + cn	T(n) = T(n/2) + cn	b
5	From the following given tree, what is the computed codeword for 'c'?		_011	_010	_100	_101	a

This sheet is for 1 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
	Unit III						
1	Fractional knapsack problem is also known as _____		0/1 knapsack problem	Continuous knapsack problem	Divisible knapsack problem	Non continuous knapsack problem	b
2	Fractional knapsack problem is solved most efficiently by which of the following algorithm?		Divide and conquer	Dynamic programming	Greedy algorithm	Backtracking	c
3	What is the objective of the knapsack problem?		To get maximum total value in the knapsack	To get minimum total value in the knapsack	To get maximum weight in the knapsack	To get minimum weight in the knapsack	a
4	Fractional knapsack problem can be solved in time O(n).		TRUE	FALSE			a
5	The result of the fractional knapsack is greater than or equal to 0/1 knapsack.		TRUE	FALSE			a
6	The main time taking step in fractional knapsack problem is		Breaking items into fraction	Adding items into knapsack	Sorting	Looping through sorted items	c
7	Which of the following algorithms is the best approach for solving Huffman codes?		exhaustive search	greedy algorithm	brute force algorithm	divide and conquer algorithm	b
8	How many printable characters does the ASCII character set consists of?		120	128	100	98	c
9	Which bit is reserved as a parity bit in an ASCII set?		first	seventh	eighth	tenth	c
10	How many bits are needed for standard encoding if the size of the character set is X?		log X	X+1	2X	X^2	a
11	The code length does not depend on the frequency of occurrence of characters.		TRUE	FALSE			b
12	In Huffman coding, data in a tree always occur?		roots	leaves	left sub trees	right sub trees	b
13	An optimal code will always be present in a full tree		TRUE	FALSE			a
14	The type of encoding where no character code is the prefix of another character code is called?		optimal encoding	prefix encoding	frequency encoding	trie encoding	b
15	What is the running time of the Huffman encoding algorithm?		O(C)	O(log C)	O(C log C)	O(N log C)	c
16	Which of the following is/are property/properties of a dynamic programming problem?		Optimal substructure	Overlapping subproblems	Greedy approach	Both optimal substructure and overlapping subproblems	d
17	When dynamic programming is applied to a problem, it takes far less time as compared to other methods that don't take advantage of overlapping subproblems.		TRUE	FALSE			a
18	A greedy algorithm can be used to solve all the dynamic programming problems.		TRUE	FALSE			b
19	Which of the following problems is NOT solved using dynamic programming?		0/1 knapsack problem	Matrix chain multiplication problem	Edit distance problem	Fractional knapsack problem	d
20	Which of the following problems should be solved using dynamic programming?		Mergesort	Binary search	Longest common subsequence	Quicksort	c
21	The Knapsack problem is an example of _____		Greedy algorithm	2D dynamic programming	D dynamic programmin	Divide and conquer	b
22	Which of the following methods can be used to solve the Knapsack problem?		Brute force algorithm	Recursion	Dynamic programming	Brute force, Recursion and Dynamic Programming	d

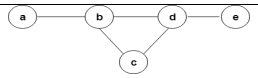
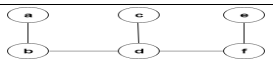
Sr. No.	Question	Image	a	b	c	d	Correct Answer
1	Given items as {value,weight} pairs {{40,20},{30,10},{20,5}}. The capacity of knapsack=20. Find the maximum value output assuming items to be divisible.		60	80	100	40	a
2	Given items as {value,weight} pairs {{60,20},{50,25},{20,5}}. The capacity of knapsack=40. Find the maximum value output assuming items to be divisible and nondivisible respectively.		100,80	110,70	1,30,110	110,80	d
3	What will be the cost of the code if character ci is at depth di and occurs at frequency fi?		cifi	∫cifi	∑fidi	fidi	c
4	From the following given tree, what is the code word for the character 'a'?		11	10	100	101	a
5	Which of the following is true? (a) h(n) is 0(f(n)) (b) h(n) is 0(g(n)) (c) g(n) is not 0(f(n)) (d) f(n) is 0(g(n))		a	b	c	d	d
6	The most common hamming codes are a generalized version of?		Hamming(7, 4) code	Hamming(8, 4) code	Hamming(6, 3) code	Hamming(5, 7) code	a

7	An Extended hamming code is also called as _____		SEDDEC	SEDDED	SECDED	SECDEC	c
8	The following sequence is a fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21,….. Which technique can be used to get the nth fibonacci term?		Recursion	Dynamic programming	A single for loop	Recursion, Dynamic Programming, For loops	d
9	Consider the recursive implementation to find the nth fibonacci number: int fibo(int n) if n <= 1 return n return _____ Which line would make the implementation complete?		fibo(n) + fibo(n)	fibo(n) + fibo(n – 1)	fibo(n – 1) + fibo(n + 1)	fibo(n – 1) + fibo(n – 2)	d
10	What is the time complexity of the following for loop method used to compute the nth fibonacci term? What is the time complexity of the following for loop method used to compute the nth fibonacci term? int fibo(int n) if n == 0 return 0 else prevFib = 0 curFib = 1 for i : 1 to n-1 nextFib = prevFib + curFib prevFib = curFib curFib = nextFib return curFib		O(1)	O(n)	O(n*n)	Exponential	b

This sheet is for 3 Mark questions							
S.r No	Question	Image	a	b	c	d	Correct Answer
Unit III							
1	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?		160	200	170	90	a
2	Consider the problem of searching an element x in an array 'arr[]' of size n. The problem can be solved in O(Logn) time if. 1) Array is sorted 2) Array is sorted and rotated by k. k is given to you and k <= n 3) Array is sorted and rotated by k. k is NOT given to you and k <= n 4) Array is not sorted		1 Only	1 & 2 only	1, 2 and 3 only	1, 2, 3 and 4	c
3	Consider the following three claims 1. (n + k)m = T(nm), where k and m are constants 2. 2n + 1 = O(2n) 3. 22n + 1 = O(2n) Which of these claims are correct ?		1 and 2	1 and 3	2 and 3	All	a
4	Consider a situation where you don't have function to calculate power (pow() function in C) and you need to calculate x^n where x can be any number and n is a positive integer. What can be the best possible time complexity of your power function?		O(n)	O(nLogn)	O(LogLogn)	O(Logn)	d
5	Maximum Subarray Sum problem is to find the subarray with maximum sum. For example, given an array {12, -13, -5, 25, -20, 30, 10}, the maximum subarray sum is 45. The naive solution for this problem is to calculate sum of all subarrays starting with every element and return the maximum of all. We can solve this using Divide and Conquer, what will be the worst case time complexity using Divide and Conquer		O(n)	O(nLogn)	O(Logn)	O(n^2)	b

This sheet is for 1 Mark questions							
S.r No	Question	Image	a	b	c	d	Correct Answer
UNIT IV							
1	The worst-case efficiency of solving a problem in polynomial time is?		O(p(n))	O(p(n log n))	O(p(n2))	O(p(m log n))	a
2	Problems that can be solved in polynomial time are known as?		intractable	tractable	decision	complete	b
3	The sum and composition of two polynomials are always polynomials.		TRUE	FALSE			a
4	_____ is the class of decision problems that can be solved by non-deterministic polynomial algorithms?		NP	P	complete	Hard	a
5	Problems that cannot be solved by any algorithm are called?		tractable problems	intractable problems	undecidable problems	decidable problems	c
6	Halting problem is an example for?		decidable problem	undecidable problems	complete	tractable problems	b
7	How many stages of procedure does a non-deterministic algorithm consist of?		1	2	3	4	b
8	A non-deterministic algorithm is said to be non-deterministic polynomial if the time-efficiency of its verification stage is polynomial		TRUE	FALSE			a
9	To which of the following class does a CNF-satisfiability problem belong?		NP class	P class	NP complete	NP hard	c
10	Which of the following problems is not NP complete?		Hamiltonian circuit	Bin packing	Partition problem	Halting problem	d

11	Which of the following algorithm can be used to solve the Hamiltonian path problem efficiently?		branch and bound	iterative improvement	divide and conquer	greedy algorithm	a
12	The problem of finding a path in a graph that visits every vertex exactly once is called?		Hamiltonian path problem	Hamiltonian cycle problem	Subset sum problem	Turnpike reconstruction problem	a
13	Hamiltonian path problem is _____		NP problem	N class problem	P class problem	NP complete problem	d
14	There is no existing relationship between a Hamiltonian path problem and Hamiltonian circuit problem.		TRUE	FALSE			b
15	Which of the following problems is similar to that of a Hamiltonian path problem?		knapsack problem	closest pair problem	travelling salesman problem	assignment problem	c
16	In graphs, in which all vertices have an odd degree, the number of Hamiltonian cycles through any fixed edge is always even		TRUE	FALSE			a
17	Which of the following case does not exist in complexity theory?		Best case	Worst case	Average case	Null case	d
18	The complexity of Binary search algorithm is		O(n)	O(log n)	O(n log n)	O(n*n)	b
19	QuickSort can be categorized into which of the following?		Brute Force technique	Divide and conquer	Greedy algorithm	Dynamic programming	b
20	The approach of dynamic programming is similar to		parcing	Hash Table	Divide and conquer	Greedy algorithm	c
21	If for an algorithm time complexity is given by O(1) then complexityof it is:		constant	polynomial	exponential	none of the mentioned	a
22	If for an algorithm time complexity is given by O(logn) then complexity will:		constant	polynomial	exponential	none of the mentioned	d

Sr. No.	Question	Image	a	b	c	d	Correct Answer
Unit IV							
1	For a graph of degree three, in what time can a Hamiltonian path be found?		$O(0.251^n)$	$O(0.401^n)$	$O(0.167^n)$	$O(0.151^n)$	a
2	What is the time complexity for finding a Hamiltonian path for a graph having N vertices (using permutation)?		O(N!)	O(N! * N)	O(log N)	O(N)	
3	How many Hamiltonian paths does the following graph have?		1	2	3	4	a
4	How many Hamiltonian paths does the following graph have?		1	2	0	3	c
5	In many problems the path to goal is irrelevant, this class of problems can be solved using _____		Informed Search Techniques	Uninformed Search Techniques	Local Search Techniques	Informed & Uninformed Search Techniques	c
6	Which of the Following problems can be modeled as CSP?		8-Puzzle problem	8-Queen problem	Map coloring problem	All	d
7	To measure Time complexity of an algorithm Big O notation is used which:		describes limiting behaviour of the function	characterises a function based on growth of function	upper bound on growth rate of the function	all of the mentioned	d

8	What is the time, space complexity of following code:
	int a = 0, b = 0;
	for (i = 0; i < N; i++) {
	a = a + rand();
	}
	for (j = 0; j < M; j++) {
	b = b + rand();
	}
			O(N * M) time, O(1) space	O(N + M) time, O(N + M) space	O(N + M) time, O(1) space	O(N * M) time, O(N + M) space	c

9	What is the time complexity of following code:
	nt a = 0;
	for (i = 0; i < N; i++) {
	for (j = N; j > i; j--) {
	a = a + i + j;
	}
	}
			O(N)	O(N*log(N))	O(N * Sqrt(N))	O(N*N)	d

10	What is the time complexity of following code:
	int a = 0, i = N;
	while (i > 0) {
	a += i;
	i /= 2;
	}
			O(N)	O(N * Sqrt(N))	O(N / 2)	O(log N)	d

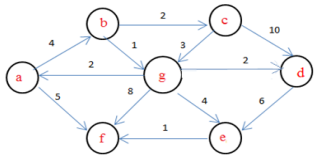
This sheet is for 3 Mark questions							
S.r No	Question	Image	a	b	c	d	Correct Answer
Unit IV							

1	What is time complexity of fun()?
	int fun(int n)
	{
	int count = 0;
	for (int i = n; i > 0; i /= 2)
	for (int j = 0; j < i; j++)
	count += 1;
	return count;
	}
			O(n^2)	O(nLogn)	O(n)	O(nLognLogn)	c

2	The time complexity of the following C function is (assume n > 0
	
	int recursive (mt n)
	{
	if (n == 1)
	return (1);
	else
	return (recursive (n-1) + recursive (n-1));
	}
			0(n)	0(nlogn)	0(n^2)	0(2^n)	d

3	Consider the matrices P, Q and R which are 10 x 20, 20 x 30 and 30 x 40 matrices respectively. What is the minimum number of multiplications required to multiply the three matrices?		18000	12000	24000	32000	a
4	What does it mean when we say that an algorithm X is asymptotically more efficient than Y?		X will always be a better choice for small inputs	X will always be a better choice for large inputs	Y will always be a better choice for small inputs	X will always be a better choice for all inputs	b

5	What is the time complexity of following code:
	int a = 0, i = N;
	while (i > 0) {
	a += i;
	i /= 2;
	}
			O(N)	O(Sqrt(N))	O(N / 2)	O(log N)	d

6	Consider the following graph., If b is the source vertex, what is the minimum cost to reach f vertex?		8	9	4	6	D
							

4	What is time complexity of fun()?
	int fun(int n)
	{
	int count = 0;
	for (int i = n; i > 0; i /= 2)
			O(n^2)	O(nLogn)	O(n)	O(nLognLogn)	c

5	The time complexity of the following C function is (assume n > 0
	
	int recursive (mt n)
	{
	if (n == 1)
	return (1);
			O(n)	O(nlogn)	O(n^2)	O(2^n)	d

3 Marks MCQ Question

S.r No	Question	Image	a	b	c	d	Correct Answer
1	Which one of the following is the recurrence equation for the worst case time complexity of theQuicksort algorithm for sorting n(≥2) numbers? In the recurrence equations given in the optionsbelow, c is a constant.	img.jpg	T(n) = 2T(n/2) + cn	T(n) = T(n−1) + T(1) + cn	T(n) = 2T(n−2) + cn	T(n) = T(n/2) + cn	b
2	Write the pseudo code for finding the factorial of given number.		int Recursive_fact(int n) { if (n==0) then return 1; else return (n*Recursive_fact(n-1)); }	{ if (n==0) then return 1; else return (n+Recursive_fact(n-1)); }	{ if (n==0) then return 1; else return (Recursive_fact(n-1)); }	none of the above is correct	a

3	How many times is the comparison i >= n performed in the following program? int i = 200, n = 80;
	main(){
	while (i >= n){
	i = i-2;
	n = n+1;
	}
	}
			20	30	42	50	c

4	Which of the following time complexities best describes MaxMin approach		Θ(n)	Ω(n)	O(nlogn)	O(logn)	a
5	Consider a set of 4 messages (M1 – M4) whose frequency of occurrences in the text is as given: (0.37, 0.51, 0.05, 0.07) Using frequency dependent Huffman Coding the codes of the messages M2 and M3 respectively.		0, 110	0, 011	1, 000	1, 001	c

This sheet is for 2 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	The characters a to h have the set of frequencies based on the first 8 Fibonacci numbers as followsa : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21/A Huffman code is used to represent the characters. What is the sequence of characters corresponding to the following code?110111100111010		**(A)** fdheg	**(B)** ecgdf	**(C)** dchfg	**(D)** fehdg	a

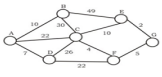
2	The basic idea behind Huffman coding is to		expand data by using fewer bits to encode more frequently occuring characters	compress data by using more bits to encode more frequently occuring characters	compress data by using fewer bits to encode fewer frequently occuring characters	compress data by using fewer bits to encode more frequently occuring characters	a
3	Which of these is false about recursion?		a) Recursive function can be replaced by a non-recursive function	b) Recursive functions usually take more memory space than non-recursive function	c) Recursive functions run faster than non-recursive function	d) Recursion makes programs easier to understand	c
4	What is tail recursion?		a) A recursive function that has two base cases	b) A function where the recursive functions leads to an infinite loop	c) A recursive function where the function doesn't return anything and just prints the values	d) A function where the recursive call is the last thing executed by the function	d
5	What happens if the base condition isn't defined in recursive programs?		a) Program gets into an infinite loop	b) Program runs once	c) Program runs n number of times where n is the argument given to the function	d) An exception is thrown	a
6	Which of these describes stepwise refinement?		a) Nicklaus Wirth described the first software engineering method as stepwise refinement	b) Stepwise refinement follows its existence from 1971	c) It is a top down approach	d) All of the mentioned	d
7	Programming based on stepwise refinement process.		a) Structural	b) C programming	c) Procedural	d) Fine	a
8	**Which of these describes stepwise refinement ?**		Nicklaus Wirth described the first software engineering method as stepwise refinement	Stepwise refinement follows its existence from 1971	It is a top down approach	All of the mentioned	d
9	Recursion is similar to which of the following?		a) Switch Case	b) Loop	c) If-else	d) if elif else	b
10	In recursion, the condition for which the function will stop calling itself is _____		a) Best case	b) Worst case	c) Base case	d) There is no such condition	c
11	Which of the following statements is true?		a) Recursion is always better than iteration	b) Recursion uses more memory compared to iteration	c) Recursion uses less memory compared to iteration	d) Iteration is always better and simpler than recursion	b
12	Consider the following instance of knapsack problem:The maximum weight of 12 is allowed in the knapsack.Find the value of maximum profit with the optimal solution of the fractional knapsack problem.	<table><tr><td>Item</td><td>X1</td><td>X2</td><td>X3</td><td>X4</td><td>X5</td></tr><tr><td>Profit</td><td>15</td><td>12</td><td>9</td><td>16</td><td>17</td></tr><tr><td>Weight</td><td>2</td><td>5</td><td>3</td><td>4</td><td>6</td></tr></table>	**(A)** 31	**(B)** 40.2	**(C)** 48.5	**(D)** None of these	**c**
13	Which of the following methods can be used to solve the Knapsack problem?		a) Brute force algorithm	b) Recursion	c) Dynamic programming	d) Brute force, Recursion and Dynamic Programming	d
14	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?		a) 160	b) 200	c) 170	d) 90	a
15	What is the time complexity of the brute force algorithm used to solve the Knapsack problem?		a) O(n)	b) O(n!)	c) $O(2^n)$	d) $O(n^3)$	c

	This sheet is for 3 Mark questions							
S.r No	**Question**	**Image**	**a**	**b**	**c**	**d**	**Correct Answer**	
1	A message is made up entirely of characters from the set X = {P,Q,R,S,T} . The table of probabilities of each character is shown below : A message of 100 characters over X is encoded using Huffman coding. Then the excepted length of the encoded message in bits is _____	<table><tr><td>Character</td><td>Probability</td></tr><tr><td>P</td><td>0.22</td></tr><tr><td>Q</td><td>0.34</td></tr><tr><td>R</td><td>0.17</td></tr><tr><td>S</td><td>0.19</td></tr><tr><td>T</td><td>0.08</td></tr><tr><td>Total</td><td>1.00</td></tr></table>	**(A)** 225	**(B)** 226	**(C)** 227	**(D)** 228	a	
2	Suppose the letters a, b, c, d, e, f have probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/32 respectively. What is the average length of Huffman codes?		**(A)** 3	**(B)** 2.1875	**(C)** 2.25	**(D)** 1.9375	**d**	
3	Given items as {value,weight} pairs {{40,20},{30,10},{20,5}}. The capacity of knapsack=20. Find the maximum value output assuming items to be divisible.		a) 60	b) 80	c) 100	d) 40	a	
4	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 nondivisible respectively.		a) 100, 80	b) 110, 70	c) 130, 110	d) 110, 80	d	
5	An alphabet consist of the letters A, B, C and D. The probability of occurrence is P(A) = 0.4, P(B) = 0.1, P(C) = 0.2 and P(D) = 0.3. The Huffman code is		A = 01 B = 111 C = 110 D = 10	A = 0 B = 11 C = 10 D = 111	A = 0 B = 111 C = 110 D = 10	A = 0 B = 111 C = 11 D = 101		

6	Consider the weights and values of items listed below. Note that there is only one unit of each item.The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by Vopt. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the	<table><tr><th>Item number</th><th>Weight (in Kgs)</th><th>Value (in Rupees)</th></tr><tr><td>1</td><td>10</td><td>60</td></tr><tr><td>2</td><td>7</td><td>28</td></tr><tr><td>3</td><td>4</td><td>20</td></tr><tr><td>4</td><td>2</td><td>24</td></tr></table>	**(A)** 16	**(B)** 8	**(C)** 44	**(D)** 60	**a**	
7	Consider the following code snippet:void my_recursive_function() { my_recursive_function(); } int main() { my_recursive_function(); return 0; } What will happen when the above snippet is executed?		a) The code will be executed successfully and no output will be generated	b) The code will be executed successfully and random output will be generated	c) The code will show a compile time error	d) The code will run for some time and stop when the stack overflows	d	
8	What is the output of the following code?void my_recursive_function(int n) { if(n == 0) return; printf("%d ",n); my_recursive_function(n-1); } int main() { my_recursive_function(10); return 0; }		a) 10	b) 1	c) 10 9 8 … 1 0	d) 10 9 8 … 1	d	
9	How many times is the recursive function called, when the following code is executed?void my_recursive_function(int n) { if(n == 0) return; printf("%d ",n); my_recursive_function(n-1); } int main() { my_recursive_function(10); return 0; }		a) 9	b) 10	c) 11	d) 12	c	
10	What will be the output of the following code?int cnt=0; void my_recursive_function(int n) { if(n == 0) return; cnt++; my_recursive_function(n/10); } int main() { my_recursive_function(123456789); printf("%d",cnt); return 0; }		a) 123456789	b) 10	c) 0	d) 9	d	

This sheet is for 3 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	Consider the given graph.What is the weight of the minimum spann		24	23	15	19	d
2	Suppose we run Dijkstra's single source shortest-path algorithm on the following edge weighted directed graph with vertex P as the source. In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized?		P, Q, R, S, T, U	P, Q, R, U, S, T	P, Q, R, U, T, S	P, Q, T, R, U, S	b
3	Consider the weights and values of items listed below. Note that there is only one unit of each item.The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by Vopt. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by Vgreedy. The value of Vopt − Vgreedy is _____ .	<table><tr><th>Item number</th><th>Weight (in Kgs)</th><th>Value (in Rupees)</th></tr><tr><td>1</td><td>10</td><td>60</td></tr><tr><td>2</td><td>7</td><td>28</td></tr><tr><td>3</td><td>4</td><td>20</td></tr><tr><td>4</td><td>2</td><td>24</td></tr></table>	16	8	44	60	a
4	What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?		Recurrence is T(n) = T(n-2) + O(n) and time complexity is O(n^2)	Recurrence is T(n) = T(n-1) + O(n) and time complexity is O(n^2)	Recurrence is T(n) = 2T(n/2) + O(n) and time complexity is O(nLogn)	Recurrence is T(n) = T(n/10) + T(9n/10) + O(n) and time complexity is O(nLogn)	b
5	Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then		T(n) <= 2T(n/5) + n	T(n) <= T(n/5) + T(4n/5) + n	T(n) <= 2T(4n/5) + n	T(n) <= 2T(n/2) + n	b
6	In a village, people build houses in the same side of the road. A thief plans to loot the village. He wants maximum amount of money without having any risk of getting caught. By some means, the villagers know that their adjacent house is being looted or not and thus they become alert. So the thief cannot loot contiguous two houses. Given that the thief knows the amount of money stored in each house and the road is straight and there is no turning, which is the most efficient algorithmic strategy to solve this problem?		Brute-force	Dynamic Programming	Backtracking	Divide and Conquer	b

7	Suppose the letters a, b, c, d, e, f have probabilities 1/2, 1/4, 1/8, 1/16, 1/32, 1/32 respectively. Which of the following is the Huffman code for the letter a, b, c, d, e, f?		0, 10, 110, 1110, 11110, 11111	11, 10, 011, 010, 001, 000	11, 10, 01, 001, 0001, 0000	110, 100, 010, 000, 001, 111	a
8	Consider a job scheduling problem with 4 jobs J_1, J_2, J_3, J_4 and with corresponding deadlines: (d_1, d_2, d_3, d_4) = (4, 2, 4, 2). Which of the following is not a feasible schedule without violating any job schedule?		J_2, J_4, J_1, J_3	J_4, J_1, J_2, J_3	J_4, J_2, J_1, J_3	J_4, J_2, J_3, J_1	b
9	consider two strings A="qpqrr" and B="pqprqrp".Let x be the length of longest common subsequence(not necessarily contiguous) between A and B and let y be no of sunch longest common subsequences between A and B. Then x+10y=		34	24	14	44	a
10	Consider the undirected graph below:Using Prim's algorithm to construct a minimum spanning tree starting with node A, which one of the following sequences of edges represents a possible order in which the edges would be added to construct the minimum spanning tree?		(E, G), (C, F), (F, G), (A, D), (A, B), (A, C)	(A, D), (A, B), (A, C), (C, F), (G, E), (F, G)	(A, B), (A, D), (D, F), (F, G), (G, E), (F, C)	(A, D), (A, B), (D, F), (F, C), (F, G), (G, E)	d

This sheet is for 2 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	which of the following is / are propert/properties of Dynamic programming	img.jpg	1 Optimal substructure	2 Overlapping subproblem	3 Greedy aproach	Both optimal substructure and overlapping substructure	d
2	prims algorith is a		Divide and conquer	Greedy method	Dynamic programming	Approzimation algorithm	b
3	Which of the following is not true about QuickSort?		a) in-place algorithm	b) pivot position can be changed	c) adaptive sorting algorithm	d) can be implemented as a stable sort	b
4	consider the following statement "A localy optimal choice is globally optimal "		True	FALSE			a
5	**Which of the following sorting algorithms has the lowest worst-case complexity?**		(A) Merge sort	(B) Bubble sort	(C) Quick sort	(D) Selection sort	a
6	Which of the following standard algorithms is not Dynamic Programming based.		Bellman–Ford Algorithm for s	Floyd Warshall Algorithm f	0-1 Knapsack problem	Prim's Minimum Spanning Tre	d
7	We use dynamic programming approach when		We need an optimal solution	The solution has optimal su	The given problem can be	It's faster than Greedy	b
8	Merge sort uses which of the following technique to implement sorting?		backtracking	b) greedy algorithm	c) divide and conquer	d) dynamic programming	c
9	Which of the following statement about 0/1 knapsack and fractional knapsack problem is correct?		In 0/1 knapsack problem iter	b) Both are the same	c) 0/1 knapsack is solved	d) In 0/1 knapsack problem ite	d
10	Which of the following is true about Huffman Coding.		Huffman coding may become lossy in some cases	Huffman Codes may not be optimal lossless codes in some cases	In Huffman coding, no code is prefix of any other code.	All of the above	b
11	which of the following statements are True statement 1 : Subgraph of shortest path is a shortest path statement 2 : If a graph contains a negative weight cycle then some shortest path may not be exist.		statement 1 is True but statement 2 is False	statement 1 is False but 2 is True	both are True	Both are False	c
12	**Let G be an undirected connected graph with distinct edge weight. Let emax be the edge with maximum weight and emin the edge with minimum weight. Which of the following statements is false?**		Every minimum spanning tree of G must contain emin	If emax is in a minimum spanning tree, then its removal must disconnect G	No minimum spanning tree contains emax	G has a unique minimum spanning tree	c
13	**Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w . Which of the following is FALSE?**		(A) There is a minimum spanning tree containing e.	(B) If e is not in a minimum spanning tree T, then in the cycle formed by adding e to T, all edges have the same weight.	(C) Every minimum spanning tree has an edge of weight w .	(D) e is present in every minimum spanning tree.	d
14	The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is _____		146	148	152	150	b
15	Match the following: List 1 (P) Prim's algorithm for minimum spanning tree (Q) Floyd-Warshall algorithm for all pairs shortest paths (R) Mergesort (S) Hamiltonian circuit List 2 (i) Backtracking (ii) Greedy method (iii) Dynamic programming (iv) Divide and conquer		P-III,Q-II,R-IV,S-I	P-I,Q-II,R-IV,S-III	P-II,Q-III,R-IV,S-I	P-II,Q-I,R-IV,S-III	b

This sheet is for 1 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	The algorithms like merge sort, quick sort and binary search are	img.jpg	1. Greedy algorithm	2. Divide and Conquer	3. Hash table	4. Parsing	b
2	QuickSort can be categorized into which of the following?		Brute Force technique	b) Divide and conquer	c) Greedy algorithm	d) Dynamic programming	b
3	Which of the following uses memoization?		Greedy approach	Divide and conquer approach	Dynamic programming approach	None of the above	c
4	What is the best case complexity of QuickSort?		a) O(nlogn)	b) O(logn)	c) O(n)	d) $O(n^2)$	a
5	The complexity of linear search algorithm is		O(n)	O(log n)	O(n2)	O(n log n)	a
6	The complexity of Binary search algorithm is		O(n)	O(log n)	O(n2)	O(n log n)	b
7	Which data structure is used for implementing a LIFO branch and bound strategy?		a) stack	b) queue	c) array	d) linked list	a
8	What are the two main features of Genetic Algorithm?		a) Fitness function & Crossov	b) Crossover techniques &	c) Individuals among the	d) Random mutation & Fitnes	a
9	What is the time complexity of Huffman Coding?		O(N)	O(NlogN)	O(N(logN)^2)	O(N^2)	b
10	Which one of the following algorithm design techniques is used in finding all pairs of shortest distances in a graph?		dynamic programming	backtracking	greedy	divide & conquer	a
11	What is the time complexity of Bellman-Ford single-source shortest path algorithm on a complete graph of n vertices?		(A) $\Theta(n^2)$	(B) $\Theta(n^2 \log n)$	(C) $\Theta(n^3)$	(D) $\Theta(n^3 \log n)$	c
12	which of the following is the most commonly used data structure for implementing Dijkstras algorithm.		max priority queue	stack	circular queue	min priority queue	d
13	What is the worst case complexity of QuickSort?		a) O(nlogn)	b) O(logn)	c) O(n)	d) $O(n^2)$	d
14	how many times the for loop gets executed in Belman ford algorithm		V times	V-1 times	E times	E-1 Times	b
15	What is the time complexity of the above dynamic programming implementation of the longest common subsequence problem where length of one string is "m" and the length of the other string is "n"?		A. O(n)	B. O(m)	C. O(m + n)	D. O(mn)	d

This sheet is for 3 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	The time complexity of the following C function is (assume n > 0) int recursive (int n) { if(n == 1) return (1); else return (recursive (n-1) + recursive (n-1)); }	img.jpg	O(n)	O(n log n)	$O(n^2)$	$O(2^n)$	d
2	The running time of an algorithm is given by Then what should be the relation between T(1), T(2) and T(3), so that the order of the algorithm is constant?		T(1) = T(2) = T(3)	T(1) – T(3) = T(2)	T(1) + T(3) = 2 * T(2)	T(1) + T(3) = 2 * T(2)	a
3	Time complexity of an algorithm T(n), where n is the input size is given by T(n)=T(n-1)+1/n, if n>1 =1, otherwise The order of this algorithm is		Logn	N	N^2	N^n	a
4	An algorithm is made up of two modules M1 and M2. If order of M1 is f(n) and M2 is g(n) then the order of algorithm is		Max(f(n),g(n))	Min(f(n),g(n))	F(n)+g(n)	F(n) * g(n)	a
5	Two alternative packages A and B are available for processing a database having 10k records. Package A requires $0.0001n^2$ time units and package B requires $10n\log_{10}n$ time units to process n records. What is the smallest value of k for which package B will be preferred over A?		12	10	6	5	c
6	The running time of an algorithm T(n),where 'n' is the input size, is given by T(n)=8T(n/2)+qn, if n>1 =p, if n=1 Where p,q are constants. The order of this algorithm is		N^2	N^n	N^3	N	c
7	Consider the following three claims 1. (n + k)m = Θ(nm), where k and m are constants 2. 2n + 1 = O(2n) 3. 22n + 1 = O(2n) Which of these claims are correct ?		**(A)** 1 and 2	**(B)** 1 and 3	**(C)** 2 and 3	**(D)** 1, 2, and 3	a
8	which one of the following correctly determines the solution of the recurrence relation with T(1)=1 T(n)=2T(n/2)+ log n		Θ(n)	Θ(nlog n)	Θ(n2)	Θ(log n)	a

9	Let X be a problem that belongs to the class NP. Then which one of the following is TRUE?		Let X be a problem that belor	If X can be solved determir	If X is NP-hard, then it is N	X may be undecidable.	c
10	Which of the following statements are TRUE? (1) The problem of determining whether there exists a cycle in an undirected gra		1, 2 and 3	1 and 3	2 and 3	1 and 2	a

This sheet is for 2 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	**What is the time, space complexity of following code:** **int a = 0, b = 0;** **for (i = 0; i < N; i++) {** **a = a + rand(); }** **for (j = 0; j < M;** **j++) { b = b + rand();** **}**	img.jpg	1. O(N * M) time, O(1) sp	2. O(N + M) time, O(N	3. O(N + M) time, O(	4. O(N * M) time, O(N + M	c
2	How many Hamiltonian paths does the following graph have?		1	2	3	4	a
3	The second smallest of n elements can be found with _____ comparisons in worst case.		N + ceil(lg n) -2	N-1	Lg n	3n/1	a
4	In general, in a recursive and non-recursive implementation of a problem (program) :		Both time and space complex	Both time and space comp	Time complexity is better	Space complexity is better in	c
5	The solution of the recurrence relation T(m)=T(3m/4)+1 is :		Θ(lg m)	Θ(m)	Θ(mlg m)	Θ(lglg m)	a
6	If there is in NP-Complete language L whose complement is in NP, then complement of any language in NP is in		P	NP	Both (A) and (B)	None of these	b
7	Of the following sorting algorithms, which has a running time that is least dependent on the initial ordering of the input?		merge sort	insrtion sort	selection sort	quick sort	a
8	The running time of quick sort algorithm depends heavily on the selection of:		No. of inputs	Arrangement of elements	Size of elements	Pivot element	d
9	The problems 3-SAT and 2-SAT are		Both NP-complete	Both in P	NP-complete and in P, re	Undecidable and NP-complete	c
10	Is it possible for a problem to be in both P and NP?		TRUE	FALSE			a
11	Indicate constant time complexity in terms of Big-O notation		O(n)	O(1)	O(log n)	$o(n^2)$	b
12	Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to		R is NP-complete	R is NP-hard	Q is NP-complete	Q is NP-hard	b
13	What is the time complexity of adding an item in front of a LinkedList		O(log n)	O(1)	$o(n^2)$	$O(2^n)$	b
14	What is the time complexity of recursive Binary Search algorithm?		O(n)	$O(2^n)$	O(log n)	O(n log n)	c
15	Assuming P != NP, which of the following is true ?		(A) NP-complete = NP	(B) NP-complete and P=∅	(C) NP-hard = NP	(D) P = NP-complete	

This sheet is for 1 Mark questions

S.r No	Question	Image	a	b	c	d	Correct Answer
1	The upper bound for the growth of the Algorithms running time is	img.jpg	Big Oh (O)	2. Big Omega (Ω)	3. Big Theta (Θ)	4. Exponential growth	a
2	Big Theta (Θ) indicates that the Upper and Lower bounds of an algorithm are the same.	TRUE	FALSE				a
3	An algorithm that breaks a file to be sorted in smaller files called runs which are sorted and eventually put back together resulting in a sorted file is called:		Quicksort algorithm	b. Replacement sort algorit	c. An indexed key sort alg	d. Mergesort algorithm	d
4	The lower bound for the growth of the Algorithms running time is represented by		Big Oh (O)	2. Big Omega (Ω)	3. Big Theta (Θ)	4. Exponential growth	b
5	Asymptotic Algorithm Analysis is primarily concerned with:		a. The size of the constant in	b. The speed of the compu	c. The speed of the comp	d. The growth rate demonstra	d
	Consider the following three claims 1. $(n + k)^m = \Theta(n^m)$, where k and m are constants 2. $2^{n+1} = O(2^n)$ 3. $2^{2n+1} =$						
6	$O(2^n)$		1 and 2	**(B)** 1 and 3	**(C)** 2 and 3	**(D)** 1, 2, and 3	
7	Problems that can be solved in polynomial time are known as?		a) intractable	b) tractable	c) decision	d) complete	b
8	_____ is the class of decision problems that can be solved by non-deterministic polynomial algorithms?		a) NP	b) P	c) Hard	d) Complete	a
9	Problems that cannot be solved by any algorithm are called?		a) tractable problems	b) intractable problems	c) undecidable problems	d) decidable problems	c
10	Halting problem is an example for?		a) decidable problem	b) undecidable problem	c) complete problem	d) trackable problem	b
11	To which of the following class does a CNF-satisfiability problem belong?		a) NP class	b) P class	c) NP complete	d) NP hard	c
12	What is vertex coloring of a graph?		a) A condition where any two	b) A condition where any t	c) A condition where all v	d) A condition where all vertic	a
13	Minimum number of unique colors required for vertex coloring of a graph is called?		a) vertex matching	b) chromatic index	c) chromatic number	d) color number	c
14	Vertex coloring and chromatic number are one and the same.		a) True	b) False			b
15	The concept of order Big O is important because:		It can be used to decide the b	It is the lower bound of the	It determines the maximu	Both (A) and (B)	a
16	The Statement "Fibonacci heap has better amortized running time in compare to a binomial heap".		a) True	b) False			a
19	A randomized algorithm uses random bits as input inorder to achieve a _____ good performance over all possible cho		a) worst case	b) best case	c) average case	d) none of the mentioned	c
20	Unix sort command uses _____ as its sorting technique.		a) Quick Sort	b) Bucket Sort	c) Radix Sort	d) Merge Sort	a

S.r No	Question	Image	a	b	c	d	Correct Answer

1	What is time complexity of fun()?
	int fun(int n)
	{
	int count = 0;
	for (int i = n; i > 0; i /= 2)
	for (int j = 0; j < i; j++)
	count += 1;
	return count;
	}
			O(n^2)	O(nLogn)	O(n)	O(nLognLogn)	c

2	The time complexity of the following C function is (assume n > 0
	
	int recursive (mt n)
	{
	if (n == 1)
	return (1);
	else
	return (recursive (n-1) + recursive (n-1));
	}
			0(n)	0(nlogn)	0(n^2)	0(2^n)	d

3	Consider the matrices P, Q and R which are 10 x 20, 20 x 30 and 30 x 40 matrices respectively. What is the minimum number of multiplications required to multiply the three matrices?		18000	12000	24000	32000	a
4	What does it mean when we say that an algorithm X is asymptotically more efficient than Y?		X will always be a better choice for small inputs	X will always be a better choice for large inputs	Y will always be a better choice for small inputs	X will always be a better choice for all inputs	b

5	What is the time complexity of following code:
	int a = 0, i = N;
	while (i > 0) {
	a += i;
	i /= 2;
	}
			O(N)	O(Sqrt(N))	O(N / 2)	O(log N)	d