USN | | | | | | | | | |

Solution of Internal Assessment Test I – May. 2021

Sub:	Design & Analysis of Algorithms					Sub Code:	18CS42	Branch:	CSE
Date:	20/05/2021	Duration:	60 min's	Max Marks:	50	Sem/Sec:	4/A,B,C & D		OBE

	Section I True or False, Multiple Choice **1 X 15 = 15**	MARKS	CO	RBT
1	If a linked list is used instead of an array in the design of the iterative algorithm to print Fibonacci sequence, it would be intractable. a) True b) False **Correct answer:** False.	1	CO1	L2
2	Consider the following procedure SQUARE which is designed to return the square of a number: INPUT: N OUTPUT: N^2 Read N from the user Return N as N * N * N Then SQUARE is an algorithm. a)True b)False **Correct answer:** True.	1	CO1	L2
3	If the complexity of an algorithm is (n) then one of the expressions could be the exact expression for the time complexity of the algorithm: a)60n b)6n + lg n c)n^2 - n d)n^2/2 **Correct answer:** a.	2	CO1	L3
4	If T(n) is O(n)and (n), then it is also: a)Theta(n) b)O(n^2) c)Theta(n^2) d Theta(1) **Correct answer:** a.	1	CO1	L3
5	Suppose the probability that an element occurs in the last position of an array is 100%, then the complexity of linear search would be: a)O(1) b)O(n) c)O(n^2) d)O(n^3) **Correct answer:** b.	2	CO1	L3

6	Suppose T(n) is O(g(n)) and Limn --> T(n) / g(n) would be equal to: a)aconstant>0 b)infinity c)0 d)variable **Correct answer:** a.	1	CO2	L3
7	Suppose T(n) = 2T(n/2) + O(nd) and T(n) is O(n lg n). Then, d is ___: a)0 b)1 c)2 d)3 **Correct answer:** b.	2	CO2	L3
8	If n < 0, why doesn't the procedure to compute factorial cease to be an algorithm? a)It computes incorrect value b)It doesn't terminate c)Leads to exponential complexity d)All the above **Correct answer:** b.	1	CO1	L2
9	The complexity of finding the maximum in a linked list of integers is a)Theta(1) b)O(1) c)Theta(n) d)O(n) **Correct answer:** d.	1	CO2	L3
10	A procedure doesn't have to be ____________ to be called an algorithm a)terminating b)effective c)correct d)producing output **Correct answer:** c.	1	CO1	L1
11	The complexity of finding if a vertex is connected to all other vertices in a graph with n vertices, when the graph is represented as an adjacency matrix is a) O(lg n) b) O(1) c) O(n) d) O(n^2) **Correct answer: c**	1	CO1	L2
12	The recurrence relation T(n) = T(n/2) + 5 is ___________ a) O(5) b) O(n/2) c) O(lg n) d) Theta(lg n) **Correct answer : c.**	2	CO2	L3

13	Problems that have feasible solutions and can be computed using algorithms within a reasonable time are known as____________ a)Intractable Problems b)Tractable Problems c)Maximization Problem d)Minimization Problem **Correct answer : b)**	1	CO1	L1
14	In a full binary tree, the number of nodes can be either even or odd. a)True b)False **Correct answer:** False**.**	1	CO3	L1

15	for i = 1 to n do
	for j = 1 to i do
	a = b * c;
	If T(n) is the time taken to execute this loop, then T(n) is O(___)
	a)n
	b)n^2
	c)n^3
	d)1
	
	Correct answer : b)
		2	CO2	L3

	Section II Short Answer Questions **4 X 5 = 20**			
1	Suppose you need to sort following key-value pairs in the increasing order of keys: INPUT: (14,5), (13, 2) (14, 3) (15,4) (16,4) Now, there are two possible solutions for the two pairs where the key is the same i.e. (14,5) and (14,3) as shown below: OUTPUT1: (13, 2), (14, 5), (14,3), (15,4), (16,4) OUTPUT2: (13, 2), (14, 3), (14,5), (15,4), (16,4) Which output is stable sort? OUTPUT1 or OUTPUT2 --------------------------(2M) Explain why.--(3M) **Correct answer: OUTPUT1** The sorting algorithm which will produce the first output will be known as a stable sorting algorithm because the *original order of equal keys are maintained.* (14, 5) comes before (14,3) in the sorted order, which was the original order i.e. in the given input, (14, 5) comes before (14,3) . On the other hand, the algorithm which produces a second output will be known as an unstable sorting algorithm because the order of objects with the same key is not maintained in the sorted order. In the second output, the (14,3) comes before (14,5) which was not the case in the original input.	5	CO1	L4

2	Without using the Master Theorem, derive the complexity in Big-O notation for the expression: T(n) = T(n/2) + n/2. Solution for the expression: T(n) = T(n/2) + n/2……………………………..(5M) **Solution:** T(n)=(T(n/4)+n/4)+n/2 => T(n)= T(n/2^2)+n/4+n/2 T(n)=(T(n/8)+n/8)+n/4+n/2 => T(n)=T(n/2^3)+n/2^3+2/4+n/2 . . . . T(n/2^i)+n/2^i+n/2^i-1+.....n/2 Let n/2^i=1 =>i=logn T(n)= T(1)+n(i/2^i+½^i-1+....½) T(n)= 1+n(1) T(n)=O(n) **Correct answer:** O (n)	5	CO1	L3
3	Let T(n) = 3n^3 + 2n^2 + 3 for an algorithm. Derive the complexity of the algorithm either using the formal definition of Big-O or using L'Hopital's rule. Solution using L'Hopital's rule or back substitution method……………………(5M) **Solution:** Back substitution Method: T(n)=3n^3 + 2n^2 + 3 g(n)=n^3 T(n)<=c*g(n) for all n>=n0 Let c=8 and n0=1 => 3n^3 + 2n^2 + 3 <=c*n^3 Hence T(n)=O(g(n)) => T(n)=O(n^3)	5	CO2	L3

4	Use the recursive and iterative algorithms to print the 10th element of the Fibonacci series and explain which of them is better by comparing the number of operations performed. To print the 10th element of the Fibonacci series…………………..(2M) Explain which of them is better by comparing the number of operations performed…………………………………………………………(3M) Solution: Recursive algorithm: fibonacci(n) If n<=1 return n else fibonacci(n-1)+fibonacci(n-2) Iterative Algorithm: Fibonacci(10) f[0]=0;f[1]=1 For i=2 to 10 f[i]=f[i-1]+f[i-2] Return f[10] Recursive version computes the element repeatedly.but in iterative version computes the elements once and stored in an array.hence it uses extra space to compute 10th element. The time complexity will be O(n) -Linear. Since the recursive function computes the intermediate terms repeatedly, the time complexity will be O(2^10) ,which is exponential.	5	CO3	L4
Section III Long Answer Questions		**1 X 10 = 10**		

1	Consider the following algorithm:
	Algorithm Mystery(n)
	//Input: A non negative integer n
	s <-- 0
	for i<-- 1 to n do
	s<--s+i*i
	return s
	What does this algorithm compute? …………………………….(2M)
	
	What is the input parameter?..(2M)
		10	CO2	L3

	If multiplication is considered to be the basic operation, how many times is the basic operation executed in terms of the input parameter? ………..(2M) Derive the time complexity expression of this algorithm and state it in the Big-O notation. I.e., O(n), O(n2) etc. Show all your work. …………….(2M) Suggest an improvement, or a better algorithm altogether, and indicate its time complexity (aka. efficiency) class (constant, logarithmic, quadratic etc.). If you cannot do it, try to prove that, in fact, it cannot be done………………….(2M) **Solution:** a. What does this algorithm compute? Ans: Computes sum of squares of n numbers b. What is the input parameter? Ans: n c. If multiplication is considered to be the basic operation, how many times is the basic operation executed in terms of the input parameter? Ans: n times d. Derive the time complexity expression of this algorithm and state it in the Big-O notation. I.e., O(n), $O(n^2)$ etc. Show all your work. Ans: ∑ 1 i=1<=n = O(n) e. Suggest an improvement, or a better algorithm altogether, and indicate its time complexity (aka. efficiency) class (constant, logarithmic, quadratic etc.). If you cannot do it, try to prove that, in fact, it cannot be done. Ans: Sum of squares of n can be computed using formula : [n(n+1)(2n+1)]/6 The time complexity is O(1).Hence the time complexity class is **Constant**			
2	Present an algorithm that searches an unsorted array a[1:n] for an element X. If X occurs twice or more in the array return true; else return false. Also analyze the complexity of the algorithm in the best, average, worst cases and denote in the order notation (Big-O) Algorithm that searches an unsorted array a[1:n] for an element X…………(3M)	10	CO1	L4

	Analyze the complexity of the algorithm in the best, average, worst cases and denote in the order notation (Big-O)…………………………………………(7M)
	
	Solution:
	Algorithm SearchX(a[1:n],X)
	Input: An array a[1:n], search key- X
	Output : true / false
	
	countX=0,i=1
	While i<=n do
	if a[i]==X then
	countX=countX+1
	if countX>=2 then
	break
	i=i+1
	return true
	else
	return false
	Analysis :
	Best- case
	If duplicate elements are present in the beginning of the array, them the number of comparison will be 2---O(1) -Constant
	worst -case:
	If the duplicate elements present at the end of the array, then a maximum number of comparisons must be done. O(n) - Linear
	Avg-case:
	If the elements are present at the middle, then the number of comparisons needed will be n/2--O(n) -Linear			

3	Consider the following recursive algorithm.
	ALGORITHM Q(n)
	
	// Input: A positive integer n
	
	if n = 1
	return 1;
	else
	return Q(n − 1) + 2 ∗ n − 1;
		10	CO2	L3

	Set up a recurrence relation for the number of multiplications made by this algorithm, derive a closed/condensed form and mention the complexity in Big-O notation. Set up a recurrence relation for the number of multiplications……………(4M) Derive a closed/condensed form and mention the complexity in Big-O notation……………………………………………………………………….(6M) **ALGORITHM Q(n)** // **Input**: A positive integer **n** **if n = 1** **return 1;** **else** **return Q(n − 1) + 2 ∗ n − 1;** Set up a recurrence relation for the number of multiplications made by this algorithm, derive a closed/condensed form and mention the complexity in Big-O notation. Solution: M(n)=M(n-1)+1, M(1)=0 M(n) = M(n-1)+1 Using backward substitution, M(n)= [M(n-2)+1]+1 = M(n-2)+2 =M(n-3)+3 ….. =M(n-k) +k When k=n-1, M(n)= M(1)+n-1 =0+n-1 =n-1 Hence M(n) belongs to O(n)			