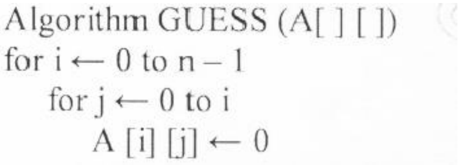
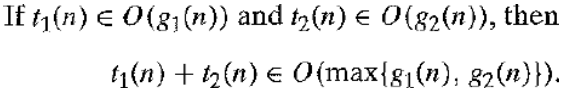
Module-1: Introduction to Algorithms

Review Questions

Introduction			
1	Define algorithm. Discuss the criteria's that an algorithm must satisfy with an example.	6	Jan 18, Jul 18, Jan19
2	Define best case, worst case and average case efficiency. Write the algorithm and give these efficiencies for sequential search.	8	Jan 20
3	Explain space complexity and time complexity with an example.	4	Jun17
4	Explain with an example how a new variable count introduced in a program can be used to find the number of steps needed by a pr ogram to solve a particular problem instance.	4	Jul 18
5	Consider the following algorithm.  Algorithm GUESS (A[ ] [ ]) for i $\leftarrow$ 0 to n – 1 for j $\leftarrow$ 0 to i A [i] [j] $\leftarrow$ 0 i) What does the algorithm compute? ii) What is basic operation? iii) What is the efficiency of this algorithm?		
Asymptotic Notations			
6	Explain asymptotic notations Big O, Big Ω and Big θ that are used to compare the order of growth of an algorithm with example.	6	Jul 17, Jul 18, Jan 19, Jul 19, Jan 20
7	Describe various basic efficiency classes.	8	Jul 19
8	Prove the following statements. a. $n^2 + 5n + 7 = \Theta(n^2)$ b. ½ $n(n-1) = \Theta(n^2)$ c. ½ $n^2+3n= \Theta(n^2)$ d. 100n + 5 =`O(n2) e. $n^2 + n = O(n^3)$ f. $5n^2 + 3n + 20 = O(n^2)$ g. $n^3 + 4n^2 = \Omega(n^2)$	6	
9	Define Little Oh. Compare the orders of growth of following functions i) (½) n (n-1) and n^2 ii) 3n+2 and n^2	6	
10	Prove that  If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, then $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$.	6	Jan 18, Jan 19. Jan 20

Mathematical Analysis of Non-Recursive Algorithms			
11	Explain general plan of mathematical analysis of **non-recursive** algorithms with example.	8	Jul 17 Jul 19
12	Write the algorithm to find **maximum element** in the given array and explain the mathematical analysis of this non-recursive algorithm.	6	Jul 18, Jul 19,
13	Write the algorithm to check whether all the elements in the given array are **distinct**and explain the mathematical analysis of this non-recursive algorithm. Derive its worst-case time complexity	6	Jan 18, Jul 19
14	Write the algorithm to perform **matrix multiplication** and explain the mathematical analysis of this non-recursive algorithm	6	
Mathematical Analysis of Recursive Algorithms			
15	Explain general plan of mathematical analysis of **recursive algorithms** with example.	8	Jan 19
16	Illustrate mathematical analysis of recursive algorithm for **Towers of Hanoi** OR Give the recursive algorithm to solve Tower of Hanoi problem. Show that the efficiency of this algorithm is exponential	8 6	Jul 17, Jul 19, Jan 20
17	Illustrate mathematical analysis of recursive algorithmto find the **factorial** of a given number.	6	Jan 19
18	State the recursive algorithm to count the **bits of a decimal number** in its binary representation. Give its mathematical analysis.	6	
19	Write a recursive function to find and print all possible **permutations** of a given set of n elements	5	Jul 18
20	Solve the recurrence relation M(n) = 2M(n-1) +1 for n>1; M(1)=1	5	Jul 18
Brute force design technique			
21			
22			
23			
24			
25			