1. Use handy summation rules to find out T(n) for the below algorithm

c1 Create an array A such that length[A] = length[X]

c2 **for** **j**← 1 **to** length[X]

c3 **do** a ← 0

c4 **for** i ← 1 **to** **j**

c5 **do** a ← a + X[**i**]

c6 A[**j**] ← a / **j**

c7 **return** A

1. Complete the Table given below for the growth rates of the funtions in the analysis of the algorithms.

|  |  |
| --- | --- |
| **Functions** | **Name** |
| **c** |  |
| **Log n** |  |
| **log2n** |  |
| **n** |  |
| **n2** |  |
| **2n** |  |

1. Inorder traversal algorithm is as shown below. What is the inorder traversal result for the given binary tree?

**inorder(tree T)**

**1: if not empty(T) then**

**2: inorder(T.left);**

**3: print(T.da);**

**4: inorder(T.right)**

**5: end if**

1. For the greatest common divisor by using of Euclid’s algorithm given below write down the Java or C function of the recursive formulation and show the recursive calls and termination conditions on the function. (10 points)

GCD(m,n)=

If n > m

If n = 0

If n > 0

1. Working mechanism of the stack is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. Working mechanism of the circular queue is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
3. The time complexity of an insertion sort or bubble sort is: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
4. Solve the following question.
	1. Trace the following algorithm with n=4.
	2. Explain what type of algorithm is this.
	3. Show recursive call and termination condition on the given algorithm.



1. Use the logic of the merge sort to sort the below given data set? What is the cost of the growing function?

8 6 4 9 7 3 6 10 12

1. For each function ***f(n)*** and time ***t*** in the following table, determine the largest size ***n*** of a problem that can be solved in time ***t***, assuming that the algorithm to solve the problem takes ***f(n)*** centiseconds (cs).(100 cs is = 1 sec

|  |  |  |
| --- | --- | --- |
| **Function****f(n)** |  **1****Second** |  **2****Days** |
| **n log n** |  |  |
|  |  |  |
| **n2** |  |  |