CS 310 'Algorithms' Fall 2019 **Quiz-3B** [Total marks=8] [Time: 20 mins]

Name: _____ Roll No: _____

Q1. Following is a description of the 'Counting Inversions' problem. You are given a sequence of n numbers $a_1, a_2, \ldots, a_n$. Assume that all the numbers are distinct. Two indices i < j form an inversion if $a_i > a_j$. Determine the total number of inversions in the input sequence.

Following is an **incomplete** divide and conquer pseudocode for counting inversions. **You have to fill in the missing statements in the code. You are not allowed to write more than one statement in each blank.** **[8 marks]**

```
Sort-and-Count(L)
  If the list has one element then
      there are no inversions
  Else
      Divide the list into two halves:
         A contains the first ⌈n/2⌉ elements
         B contains the remaining ⌊n/2⌋ elements
      (r_A, A) = Sort-and-Count(A)
      (r_B, B) = Sort-and-Count(B)
      (r, L) = Merge-and-Count(A, B)
   Endif
   Return r = r_A + r_B + r, and the sorted list L
```

Merge-and-Count(A, B)	
1	Maintain a Current pointer into each list, initialized to point to the **front elements**
2	Maintain a variable Count for the number of inversions, initialized to 0
3	**While** both lists are ___**nonempty**_____
4	Let a_i and b_j be the elements pointed to by the Current pointer
5	Append to the output list: ____**the smaller of these two elements**_____
6	**If** (**b_j is the smaller element**) then
7	**Increment count by the number of elements remaining in A**
9	**Endif**
10	**Advance the current pointer in the list from which the smaller element was selected**
11	**EndWhile**
12	Once one list is empty, **append the remainder of the other list to the output**
13	Return Count and the merged list

What is the recurrence relation for Sort-and-Count? **T(1) = O(1) and T(n) = 2T(n/2) +O(n) if n>1**