National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Design and Analysis of Algorithms	Course Code:	CS 302
Program:	CS	Semester:	Fall 2018
Duration:	60 Minutes	Total Marks:	35
Paper Date:		Weight	
Section:	ALL	Page(s):	1
Exam Type:	Sessional - I		

Student : Name:______________________________ Roll No.________________ Section:_______

Instruction/Notes: Solve this exam on your answer sheets .

Problem 1 [15 pts]

Let A be an array of n distinct numbers. For indices i and j, if $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A (meaning that the numbers $A[i]$ and $A[j]$ appear in A in the inverted order).

a. List all the inversions in the array {2, 3, 8, 6, 1}

b. Give an algorithm that determines the number of inversions in an array A of size n in $O(nlgn)$ worst-case time.
Write the algorithm as a bulleted list of clearly specified statements, using English and symbols where necessary.

Problem 2 [10 pts]

Perform step-count analysis on the following pieces of code. Determine equations for the time functions $T(n)$ and express them in the big-Oh notation.

a.

```
for(int i=1;i<=n; i=i*2){
     for(int j=1; j<=i; j++){
               sum++;
     }
}
```

b.

```
for(int i=1;i<=n; i=i*2){
     for(int j=n; j>=1; j=j/2){
               sum++;
     }
}
```

Problem 3 [10 points]

a. If we modify the merge sort algorithm so that instead of dividing the array into two equal parts it divides it into three equal parts, sorts them separately, and then merges them to sort the original array, how will this effect the asymptotic running time of the algorithm?

b. Two arrays, A and B, both of size n, represent two mathematical sets of n numbers each. We wish to take the intersection of these sets. What is the fastest possible time in terms of big-Oh in which this can be done? Give a precise reason for your answer.