CS4335 Design and Analysis of Algorithms
Tutorial 5

Question 1. Consider the sorting problem. Suppose that the list is:
3, 2, 1, 5, 8, 9, 10, 4, 7, 6, 12, and 11. Use the merge sort algorithm to sort the list in increasing order. Show the intermediate steps.

Question 2. For the same list: 3, 2, 1, 5, 8, 9, 10, 4, 7, 6, 12, and 11. Suppose we have sorted the two halves as list1: 1, 2, 3, 5, 8, 9; and list2: 4, 6, 7, 10, 11, 12. Calculate the number of inversions with one number in list1 and the other number in list2 using *O(n)* operations.

Question 1:

List:	3, 2, 1, 5, 8, 9, 10, 4, 7, 6, 12, 11										
Divide:	3, 2, 1, 5, 8, 9						10, 4, 7, 6, 12, 11				
Divide:	3, 2, 1			5, 8, 9			10, 4, 7			6, 12, 11	
Divide:	3, 2		1	5, 8		9	10, 4		7	6, 12	11
Divide:	3	2	1	5	8	9	10	4	7	6	12	11
Merge:	2, 3		1	5, 8		9	4, 10		7	6, 12	11
Merge:	1, 2, 3			5, 8, 9			4, 7, 10			6, 11, 12	
Merge:	1, 2, 3, 5, 8, 9						4, 6, 7, 10, 11, 12				
Merge:	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12										

Question 2:

Sorted list1: 1, 2, 3, 5, 8, 9; Sorted list2: 4, 6, 7, 10, 11, 12
Sorted: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

$$number\ of\ inversions = 3 + 2 + 2 + 0 + 0 + 0$$
$$= 7$$