# Lab 3 (24 Jan 2019)

**Problem 1 [Counting Inversions]:** An *inversion* in a sequence A of numbers is a pair of indices (i,j) such that i<j and A[i]>A[j]. Write a program to count the total number of inversions in an input sequence. For e.g. the number of inversions in 1,3,9,8,5 is 3 while that in 4,10,8,2,1 is 8. [*Write a simple naive algorithm to do this.* Write a program to implement the divide-and-conquer algorithm to count inversions.

**Problem 2 [Maximum sum sub-array]:** Given an array A of integers write a program to return indices (i,j) (0 <= i < j <= n-1) such that the sum A[i]+A[i+1]+...+A[j] is maximum for all sub-arrays. For e.g. if A = [-2, 10, -4, 12, -9], then i=1,j=3 would give the maximal sum (10+ (-4) + 12=18). Your algorithm should run in O(nlogn) time.
*Hint: Think of a divide-and-conquer strategy. You can implement a naive $O(n^2)$ algorithm to check the correctness of your divide-and-conquer algorithm. Do you think you can solve this problem in O(n) time?*