Data Structures and Algorithms

Programming Assignment #2

Topics : Divide and Conquer

Instructions for Submission

1. Register on spoj.com and a2oj.com. Link your *spoj* account to *a2oj*
2. Go to the front page of *a20j* and search for the contest titled **Crash Course (DS & Algo**. The contest code is **40614**.
3. Register for the contest. You'll see a pop-up with 5 problems. Before submitting the problems, make sure that you're logged into your spoj account. Click on any problem and it should redirect you to the spoj page. Submit your question on **this** page. After submission, your ranking should be updated withing 5 minutes.

Problems

1. **Maximum Sum Subarray** : Given an array consisting of positive and negative integers, write an algorithm to find the maximum sub array. A subarray is defined as a contiguous sequence of elements. A subarray can of be length 1 as well as of length N, but it cannot be empty. Note that you only need to return the sum and not the subarray. Start out with an $O(n^3)$ solution. Then improve this to $O(n^2)$. Code the $O(n^2)$ solution and test it and submit it. It should result in *TLE* and not *Wrong Answer*. Then improve it $O(n \log n$ and submit it. That should clear all the test cases.

   Hint : Write a function called *Max_Crossing_Subarray*.

   Follow - Up : How do you print any 1 subarray which has the maximum sum?

2. **Counting Inversions** : Given an array $A[1, ...N]$ (with possible repetitions), We say that for $1 \leq i < j \leq n$, a pair (i, j) is an inversion if $a_i > a_j$. Our goal is to count the number of inversions in an array. First, write an $O(n^2)$ algorithm and submit it. It should result in *TLE* and not *Wrong Answer* . Finally, optimize it to $O(n \log n)$.

   Hint : Think of a creative function definition. What if I told you to give me the number of inversions and also return a sorted array? Would that be an easy version of the problem. (Of course, you should not sort the array first using library functions as it would destroy the order).