CSC 413/513 Course Programming Project – Q&A

Inversion Count by Divide-and-Conquer

Question 1:

Good afternoon Dr. Li,

I am having some trouble determining the best place to count the inversions with the project. With it being a divide and conquer algorithm, would it be best to count inversions during the merge steps? Or during the sort? or both?

In my mind, it makes the most sense on the merge steps, due to the fact that you can just test for inversions at the lowest merge point for each branch and obtain an exact count. But, I have tried a few different ways, but none of them have been 100% accurate for every test case. If you have any direction, I would greatly appreciate it.

Answer:

The idea is to add the inversion counting during sorting. What you need to do is adding the inversion counting part to the MergeSort algorithm.

Below are two helpful observations for you to develop the algorithm.

Observation 1: *Given two subarrays, the inversions between the first and second subarrays remains the same regardless of the order of elements in each subarray.*

Observation 2: *Given two sorted subarrays A1 and A2, if A1[i] > A2[j], A1[i] and A2[j] are inverted, and every element in A1 after A1[i] is an inversion of A[j] as well.*