**Algorithm Design Lab 3**

**Document Id 50**

**Questions**

Q1 Define multi-branch recursion w.r.t Divide-and-Conquer Algorithms

Q2 Explain how Divide-and-Conquer Algorithms support concurrent programming.

Q3 Explain how memoization could be a negative attribute for a complex Divide-and-Conquer

Algorithm

Q4 Explain how a Dynamic solution can be superior to a Divide-and-Conquer solution

**Practical Task Count the Unique Inversions**

Modify the Divide-and-Conquer Solution to count the inversions

Your solution must be a able to handle

* Ignore repeated values
* Ignore repeated inversions

Use the following input data

Input data

**int simpleInversion[] = {6, 7, 1, 2, 3, 4, 5}; ANS 1**

**int simpleRepeatInversion[] = {2, 2, 1, 2}; ANS 1**

**int complexRepeatInversion[] = {3, 2, 3, 4, 4, 4, 4, 5, 3, 3}; ANS 3**