Thomas Steinke
CPE 349
Counting Inversions Lab

Pseudo code

```
// Input: Array of comparable items
// Output: Number of inversions in the array
CountInversionsAndSort( array[0..n-1] )
    numInversions = 0
    A = array[0.. (n-1)/2 ]
    B = array[ (n-1)/2 ..n-1]
    add CountInversionsAndSort( A ) to numInversions
    add CountInversionsAndSort( B ) to numInversions
    add MergeAndCount( A, B, => array ) to numInversions

    return numInversions

// Merge the two halves of the array together
MergeAndCount( leftHalf[0.. (n-1)/2 ], rightHalf[0.. (n-1)/2 ], => destination )
    numInversions = 0
    left = 0, right = 0
    while left < (n-1)/2 and right < (n-1)/2
        if rightHalf[right] < leftHalf[left]
            destination[left + right] = rightHalf[right]
            right = right + 1
        else
            // Since the left half has a larger number, that's (the value of right) inversions
            add right to numInversions
            destination[left + right] = leftHalf[left]
            left = left + 1
    retun numInversions
```

Recurrence Relation

$CI(n) = 2 * CI(\frac{n}{2}) + n$

According to the Master Theorem (a = 2, b = 2, d = 1):

$CI \in \theta(n \log n)$