

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..  $\frac{n-1}{2}$  ]
    B = array[  $\frac{n-1}{2}$  ..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..  $\frac{n-1}{2}$  ], rightHalf[  $\frac{n-1}{2}$  ], => destination )
    numInversions = 0
    left = 0, right = 0
    while left <  $\frac{n-1}{2}$  and right <  $\frac{n-1}{2}$ 
        if leftHalf[left] > rightHalf[right]
            // 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
        else
            destination[left + right] = rightHalf[right]
            right = right + 1
    return numInversions
```

Recurrence Relation

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

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

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