Counting Inversions

Inversion

$A[i] > A[j]$

where $i < j$

Example

7 , 6 , 3 , 8

for 7.

2 inversions.

for 6

1 inversion.

for 3

0 inversions

for 8

0 inversions as there are no elements to its right.

Total inversions = 3

Divide Step : separate list into 2 pieces

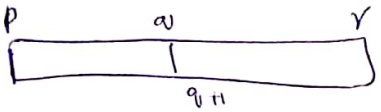
Conquer : recursively count inversions in each half.

Combine : Count inversions where a_i and a_j are in different halves and return sum of three quantities

* Assume each half is sorted
* count inversions where a_i & a_j are in different halves
* Merge 2 sorted halves into sorted whole (to maintain sorted invariant)

```
MS_In(A, p, r)
{   if n = 1
        return 0
    else
        X = MS_In(A, p, q)
        Y = MS_In(A, q+1, r)
        Z = M_In(A, p, q, r)

    Inversions = X + Y + Z
    return inversions }
```




```
Merge_Inversions(A, p, q, r)
{   n1 = q - p + 1          n2 = r - q
    L[1 ... n1]             R[1 ... n2]
    for i = 1 → n1
        L[i] = A[p + i - 1]
    for j = 1 → n2
        R[j] = A[q + j]
    i = 1 , j = 1
    for k = p to r
    { if L[i] ≤ R[j]
          A[k] = L[i], i++
      else
          A[k] = R[j], j++
          inversions += len(L) - i + 1 }
    return inversions
```