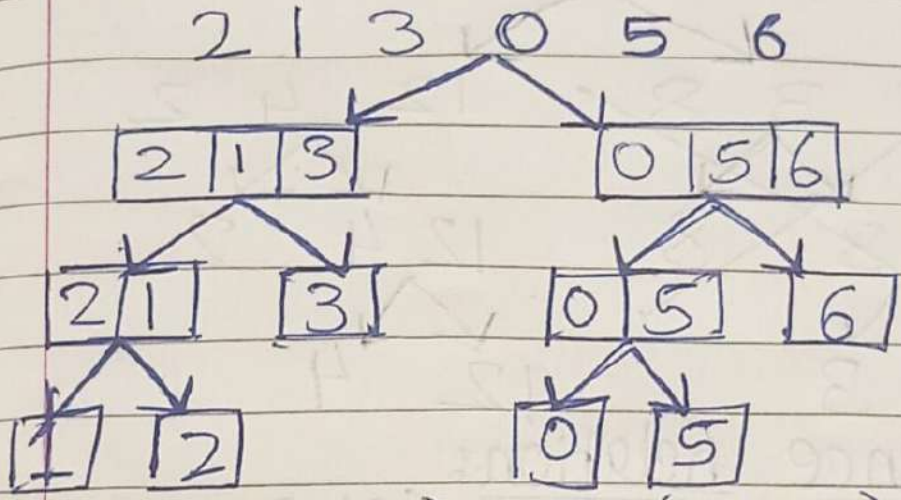
* Count - Inversion Algorithm



(1, 2) : (2, 1)

Recurrence Relation:

$T(n) = 2T(n/2) + O(n)$

~~Mer~~ $a = 2 \quad b = 2 \quad d = 1$

$a = b^d \Rightarrow O(n \log n)$

Algorithm:

- Split the given input array into two halves.
- Count the inversion in left, right and while merging the two.
- Stop when 1 element is in each half.
- while((i < mid) && (j <= right))

```
if(arr[i] <= arr[j])
    temp[k] = arr[i]
else
    temp[k] = arr[j]
```

```
while (i< mid)
  temp[k] = arr [i]
while (j<= right
  temp[k] = arr [j]
```