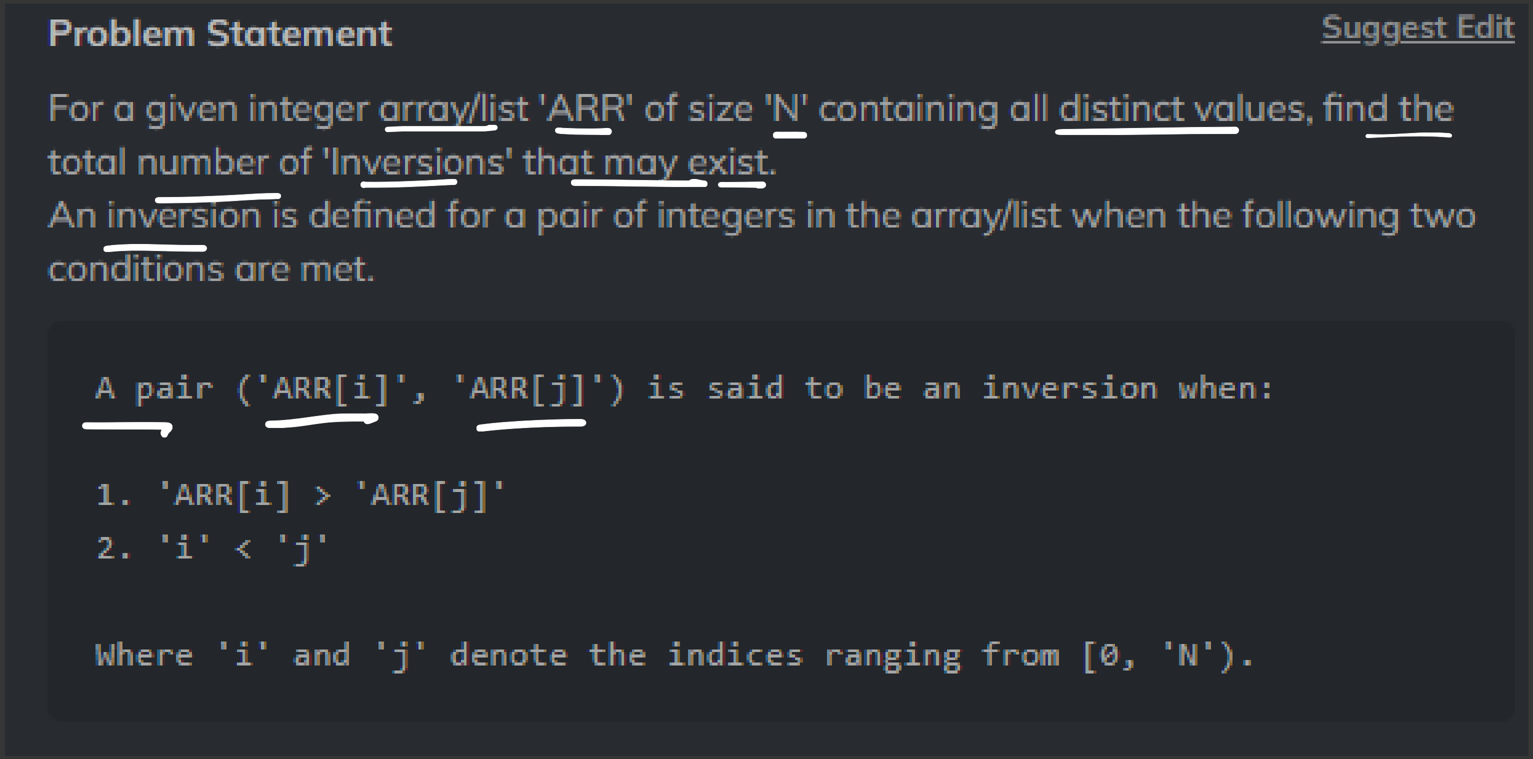
Count Inversion



Problem Statement Suggest Edit

For a given integer array/list 'ARR' of size 'N' containing all distinct values, find the total number of 'Inversions' that may exist.
An inversion is defined for a pair of integers in the array/list when the following two conditions are met.

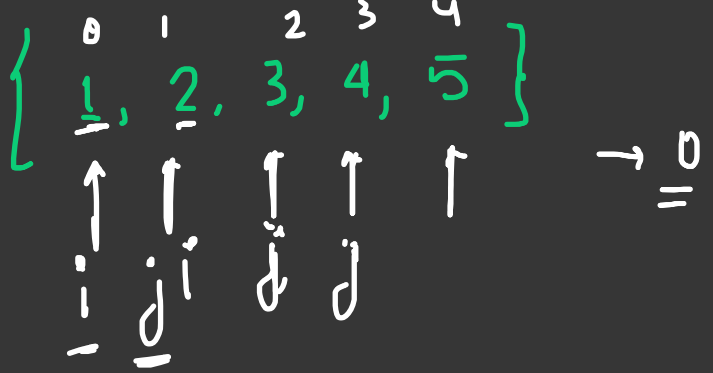
```
A pair ('ARR[i]', 'ARR[j]') is said to be an inversion when:

1. 'ARR[i] > 'ARR[j]'
2. 'i' < 'j'

Where 'i' and 'j' denote the indices ranging from [0, 'N').
```

Count Inversion $\rightarrow$ pair of integer $\{arr[i], arr[j]\}$

$arr[i] > arr[j]$

$i < j$

arr[ ] :

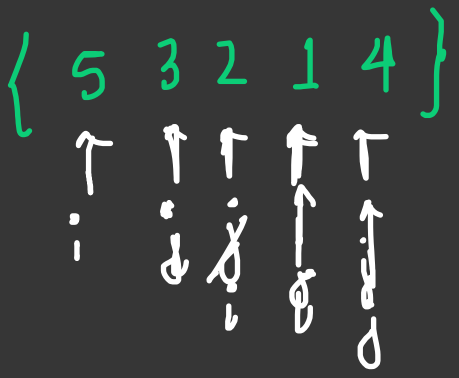


i j

arr[i] > arr[j]

count = ~~3~~ 4 ~~5~~ 6 $\bigcirc 7$

arr[ ] :

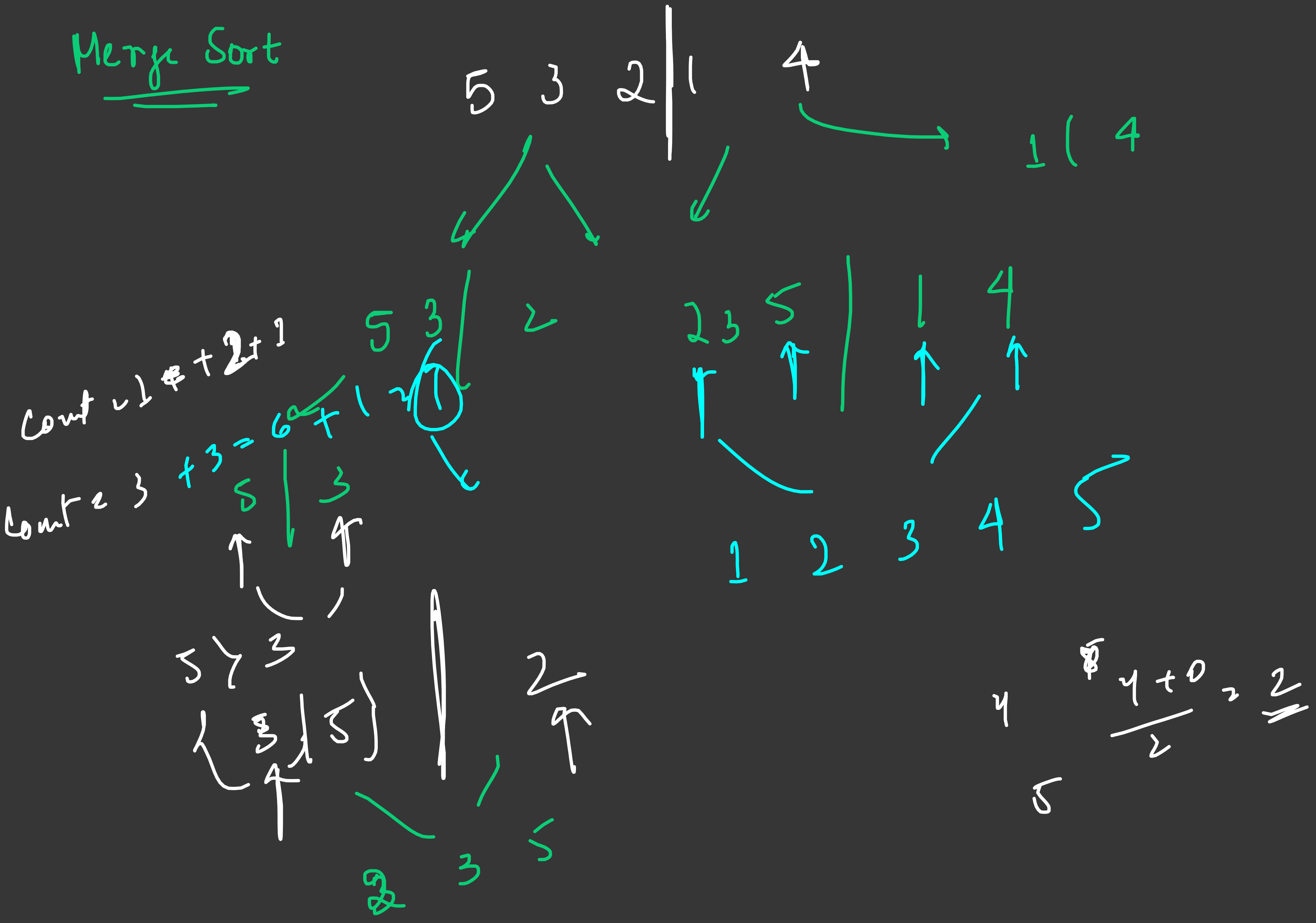


Brute force:

```
for(i=0; i<n, i++)
    for(j= i+1, j<n, j++)
        if(arr[i] > arr[j])
            count++,
```

T.C. = $O(N^2)$

Merge Sort



5 3 2 | 1 4
1 | 4
5 3 | 2
2 3 5 | 1 4
1 2 3 4 5
Count = 1 + 2 + 1
Count = 3 + 3 = 6 + 1
5 | 3
5 > 3
3 | 5
2 3 5
4 + 0 / 2 = 2
4
5

Merge Sort

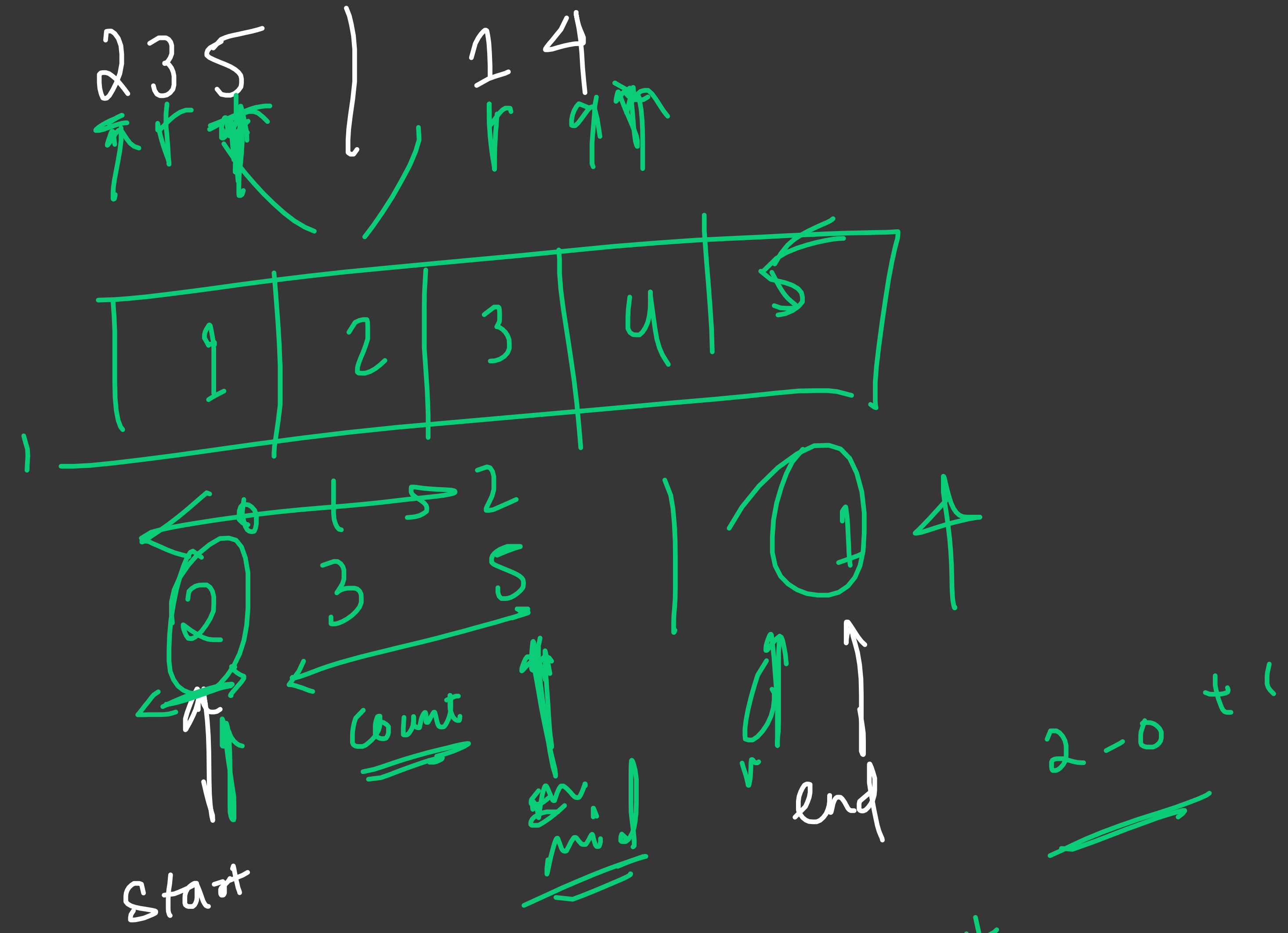
Merge {

) {

}

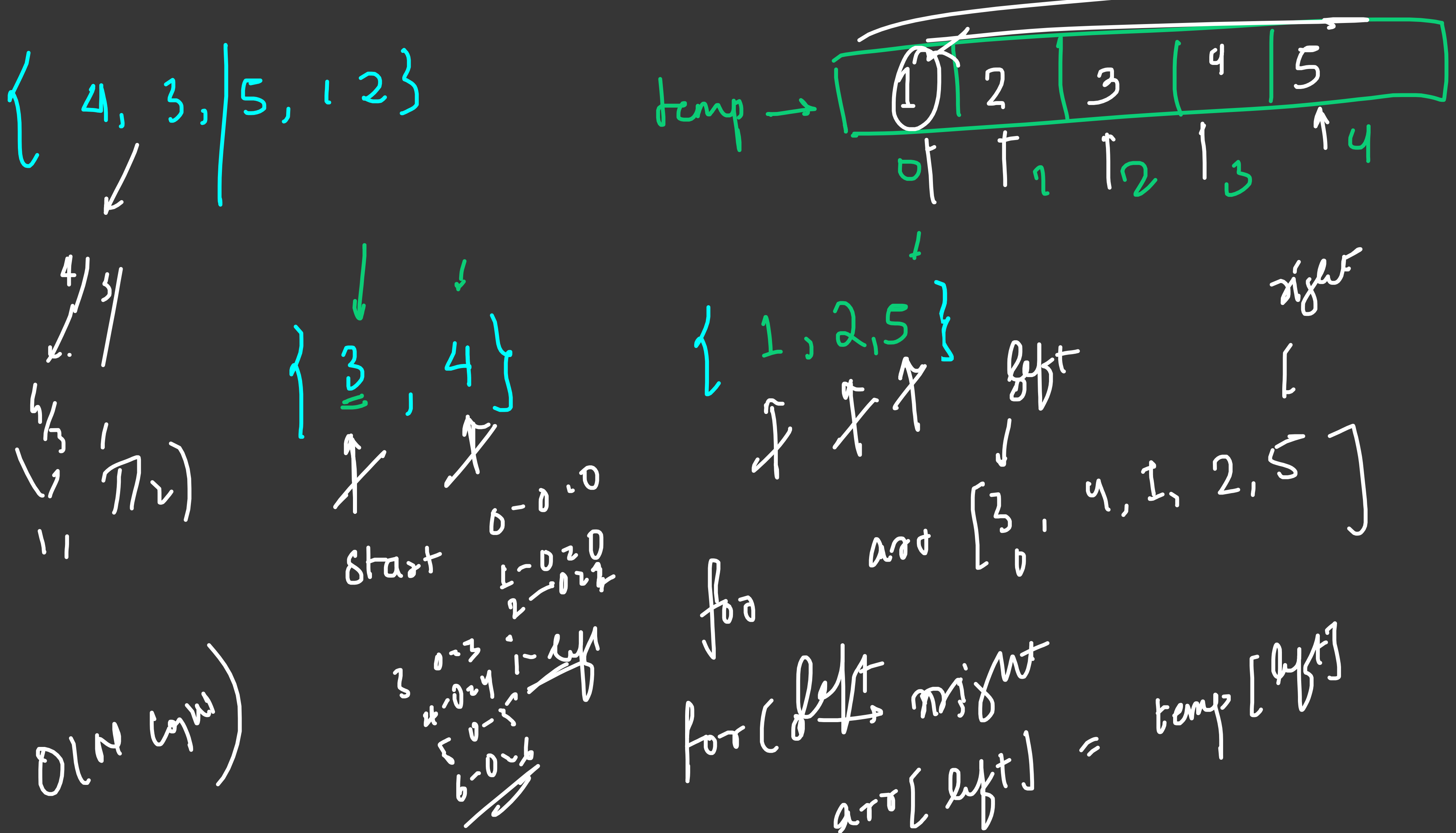
$$\frac{left + right}{2} + 1 \qquad n-1$$

```
merge Sort{ int left , mid, right ) {
    if ( left < right) {
        mergeSort( 0, mid)
        merge sort( mid+1, right)
        merge ( left, mid, right)
}
```



Count = ~~Start - mid~~ mid - start

2 - 0 3



for

for(left → right)

arr[left] = temp[left]

T.C. : $O(N\log N)$

S.C. : $O(N)$ $\longrightarrow$ as we are using temp vector for temporary sort sorted elements