```
function countInversions (arr) {
    const n = arr.length

    if (n === 1) return {sortedArr: arr, inversionCount: 0}

    const mid = Math.floor(n/2)
    const left = arr.slice(0, mid)
    const right = arr.slice(mid)

    const leftInversions = countInversions(left)
    const rightInversions = countInversions(right)
    const mergedInversions = mergeSortedArraysAndCountInversions(leftInversions.sortedArr, righ
tInversions.sortedArr)

    return {
        sortedArr: mergedInversions.sortedArr,
        inversionCount: leftInversions.inversionCount + rightInversions.inversionCount + merged
Inversions.inversionCount
    }
}

function mergeSortedArraysAndCountInversions (leftArr, rightArr) {
    let sortedArr = []
    let i = 0
    let j = 0
    let inversionCount = 0

    while (i < leftArr.length && j < rightArr.length) {
        if (leftArr[i] <= rightArr[j]) {
            sortedArr.push(leftArr[i])
            i++
        }
        else {
            sortedArr.push(rightArr[j])
            j++
            inversionCount += leftArr.length - i
        }
    }

    while (i < leftArr.length) {
        sortedArr.push(leftArr[i])
        i++
    }

    while (j < rightArr.length) {
        sortedArr.push(rightArr[j])
        j++
    }

    return {sortedArr, inversionCount}
}
```

```
const array = [1, 9, 6, 4, 5];
console.log(countInversions(array));
```