



## Merge Sort: Counting Inversions ☆

Your Merge Sort: Counting Inversions submission got 45.00 points.

Share

Tweet



[Proceed to Interview Preparation Kit](#)

### Problem

### Submissions

### Leaderboard

### Editorial

In an array,  $arr$ , the elements at indices  $i$  and  $j$  (where  $i < j$ ) form an inversion if  $arr[i] > arr[j]$ . In other words, inverted elements  $arr[i]$  and  $arr[j]$  are considered to be "out of order". To correct an inversion, we can swap adjacent elements.

For example, consider the dataset  $arr = [2, 4, 1]$ . It has two inversions:  $(4, 1)$  and  $(2, 1)$ . To sort the array, we must perform the following two swaps to correct the inversions:

$$arr = [2, 4, 1] \xrightarrow{swap(arr[1], arr[2]) \rightarrow swap(arr[0], arr[1])} [1, 2, 4]$$

Given  $d$  datasets, print the number of inversions that must be swapped to sort each dataset on a new line.

### Function Description

Complete the function `countInversions` in the editor below. It must return an integer representing the number of inversions required to sort the array.

`countInversions` has the following parameter(s):

- `arr`: an array of integers to sort.

### Input Format

The first line contains an integer,  $d$ , the number of datasets.

Each of the next  $d$  pairs of lines is as follows:

- The first line contains an integer,  $n$ , the number of elements in  $arr$ .
- The second line contains  $n$  space-separated integers,  $arr[i]$ .

### Constraints

- $1 \leq d \leq 15$
- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 10^7$

### Output Format

For each of the  $d$  datasets, return the number of inversions that must be swapped to sort the dataset.

### Sample Input

```
2
5
1 1 1 2 2
5
2 1 3 1 2
```

### Sample Output

```
0
4
```



**Explanation**

We sort the following  $d = 2$  datasets:

1.  $arr = [1, 1, 1, 2, 2]$  is already sorted, so there are no inversions for us to correct. Thus, we print **0** on a new line.

2.  $arr = [2, 1, 3, 1, 2] \xrightarrow{1 \text{ swap}} [1, 2, 3, 1, 2] \xrightarrow{2 \text{ swaps}} [1, 1, 2, 3, 2] \xrightarrow{1 \text{ swap}} [1, 1, 2, 2, 3]$

We performed a total of  $1 + 2 + 1 = 4$  swaps to correct inversions.

Change Theme

C++



```

37     }
38     while(j < n2){
39         (*arr)[k] = rightArr[j];
40         j++;
41         k++;
42     }
43     return count;
44 }
45 long countInversions_rec(vector<int> *arr, int leftBegin, int rightEnd){
46     long inversions = 0;
47     if(leftBegin >= rightEnd){
48         return 0;
49     }
50     int mid = floor( (leftBegin + rightEnd) / 2 );
51     inversions = countInversions_rec(arr, leftBegin, mid);
52     inversions += countInversions_rec(arr, mid+1, rightEnd);
53     return inversions + merge(arr, leftBegin, mid, rightEnd);
54 }
55 long countInversions(vector<int> arr) {
56     vector<int> *temp = &arr;
57     int n = ( temp->size() );
58     return countInversions_rec(temp, 0, n-1);
59 }
60 //https://www.geeksforgeeks.org/counting-inversions/
61 //https://medium.com/@carlosbf/merge-sort-counting-inversions-solution-46a0b311b20a

```

Line: 61 Col: 84

☒ Upload Code as File ☐ Test against custom input

Run Code

Submit Code

## Congratulations

You solved this challenge. Would you like to challenge your friends?

Proceed

✔ Test case 0

✔ Test case 1

✔ Test case 2

✔ Test case 3

✔ Test case 4

Compiler Message

Success

Input (stdin)

```

1 2
2 5
3 1 1 1 2 2
4 5

```

Download



✔ Test case 5 🔒

✔ Test case 6 🔒

5 | 2 1 3 1 2

Expected Output

Download

1 | 0

2 | 4

