HackerRank | Prepare Certify Compete Apply Search

All Contests > utp-open-2018 > How Many Inversions?

How Many Inversions?

Problem Submissions Leaderboard Discussions

$Humbertov\ Moralov$ in his student days, he is attending system engineering at "University of missing hill". He was evaluated in its first course of Analysis of Algorithms (at the first half of 1997) with the following topics and questions:

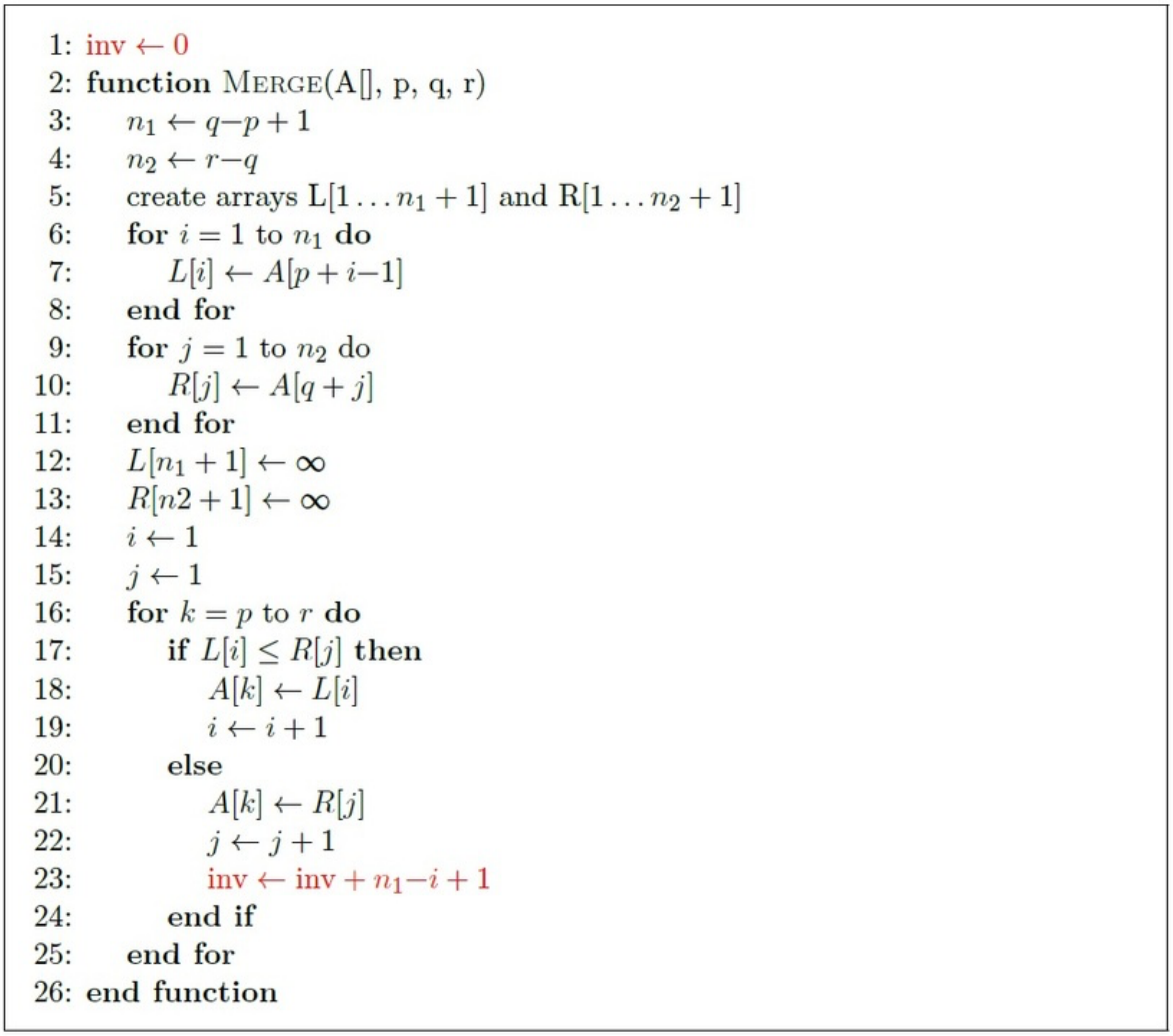
Inversions :

Let $A[1 \dots n]$ an array of distinct integers of size n. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A.

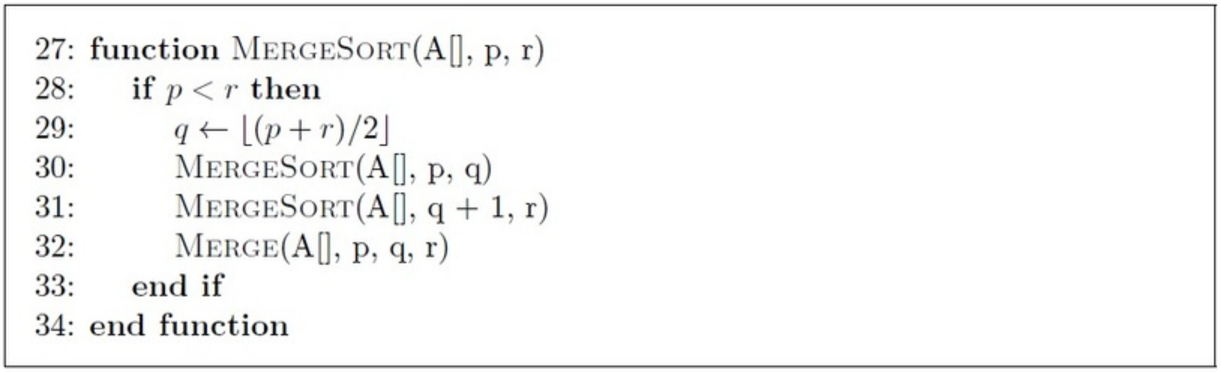
Given the above definition about an inversion, *Humbertov Moralov* must answer the following questions:

- List all inversions in $\langle 3, 2, 8, 1, 6 \rangle$.
- What array of size n, with all the numbers from the set $1, 2, 3, \dots, n$ has the largest amount of inversions?. How many inversions?.
- Write an algorithm to determine the number of inversions in any permutation of n elements with $\theta(n \log n)$ in the worst case run time.

Humbertov Moralov answered questions 1. and 2. without any problem, but he was not able to solve the question 3. at time. Days later he thought the following solution:



```
 1: inv ← 0
 2: function MERGE(A[], p, q, r)
 3:     $n_1 \leftarrow q - p + 1$
 4:     $n_2 \leftarrow r - q$
 5:     create arrays L[1 . . . $n_1 + 1$] and R[1 . . . $n_2 + 1$]
 6:     for $i = 1$ to $n_1$ do
 7:         $L[i] \leftarrow A[p + i - 1]$
 8:     end for
 9:     for $j = 1$ to $n_2$ do
10:         $R[j] \leftarrow A[q + j]$
11:     end for
12:     $L[n_1 + 1] \leftarrow \infty$
13:     $R[n2 + 1] \leftarrow \infty$
14:     $i \leftarrow 1$
15:     $j \leftarrow 1$
16:     for $k = p$ to $r$ do
17:         if $L[i] \leq R[j]$ then
18:             $A[k] \leftarrow L[i]$
19:             $i \leftarrow i + 1$
20:         else
21:             $A[k] \leftarrow R[j]$
22:             $j \leftarrow j + 1$
23:             inv ← inv + $n_1 - i + 1$
24:         end if
25:     end for
26: end function
```



```
27: function MERGESORT(A[], p, r)
28:     if $p < r$ then
29:         $q \leftarrow \lfloor (p + r)/2 \rfloor$
30:         MERGESORT(A[], p, q)
31:         MERGESORT(A[], q + 1, r)
32:         MERGE(A[], p, q, r)
33:     end if
34: end function
```

Will this code solve the problem?. Just adding the lines 1 and 23 will be enough to solve the problem?

Please help *Humbertov Moralov* to validate this solution!. For this, you must implement this solution in any of the programming languages accepted by the ICPC (International Collegiate Programming Contest) and verify if the expected results are generated.

Input Format

The first line contains an integer t $(1 \leq t \leq 10)$ denoting the number of cases, each test case has the following structure:

The first line contains a positive integer n $(1 \leq n \leq 10^6)$, which represent the length of A.

The second line contains n space-separated positive integers which make up the array A, these values are in the closed interval $[1,\ 10^8]$.

Constraints

$1 \leq t \leq 10$
$1 \leq n \leq 10^6$
$i \in [1, n]$
$1 \leq A[i] \leq 10^8$

Output Format

For each test case, your program must print a non-negative integer representing the total number of inversions in the array A. Each valid test case must generate just one output line.

Sample Input 0

```
3
5
3 2 8 1 6
5
5 4 3 2 1
1
10
```

Sample Output 0

```
5
10
0
```

Submissions: 64
Max Score: 100
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File

□ **Test against custom input**

Run Code

Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |