Yet Another Contest 8 P1 - Permutation Sorting

Time limit: 3.0s **Memory limit:** 256M

Nils likes sorted permutations. He currently has a permutation of the first N positive integers $p_1, p_2, \ldots, p_N$, and wishes to sort it. He can apply the following operation any number of times:

- Choose integers l, r satisfying $1 \leq l \leq r \leq N$. Then, sort the subarray $p_l, p_{l+1}, \ldots, p_r$, leaving the other elements unchanged. The cost of the operation is $r - l + 1$.

For example, if the permutation was $1, 5, 3, 4, 2$, and Nils chooses $l = 2$ and $r = 4$, then the permutation will become $1, 3, 4, 5, 2$, incurring a cost of 3.

What is the minimum total cost to sort the permutation?

Constraints

$1 \leq N \leq 10^6$

$p_1, p_2, \ldots, p_N$ is a permutation of the integers $1, 2, \ldots, N$, that is, every integer between 1 and N (inclusive) occurs exactly once in p.

Subtask 1 [20%]

There exists an optimal solution which applies at most one operation.

Subtask 2 [20%]

$1 \leq N \leq 2000$

Subtask 3 [60%]

No additional constraints.

Input Specification

The first line contains a single integer, N.

The second line contains N space-separated integers, $p_1, p_2, \ldots, p_N$.

Output Specification

On a single line, output the minimum total cost to sort Nils' permutation.

Sample Input

```
6
2 1 3 6 5 4
```

Sample Output

```
5
```

Explanation

It is optimal to first sort the subarray with $l = 1$ and $r = 2$. Then, the subarray with $l = 4$ and $r = 6$ can be sorted.