

CS1020E | Lab 7 | Exercise 1

Swinging Monkey

Objectives

The objective of this exercise is to learn to use a **stack** to solve a problem more efficiently.

Problem Description

Monkeys like to swing from tree to tree. They can swing from one tree to another directly as long as there is no tree in between that is taller than or have the same height as either one of the two trees. For example, if there are 5 trees with heights 19m, 17m, 20m, 20m and 20m lining up in that order, then the monkey will be able to swing from one tree to the other as shown below:

1. from first tree to second tree
2. from first tree to third tree
3. from second tree to third tree
4. from third tree to fourth tree
5. from fourth tree to fifth tree

Tarzan, the king of jungle who is able to communicate with the monkeys, wants to test the monkeys to see if they know how to count the total number of pairs of trees that they can swing directly from one to the other. But he himself is not very good in counting. So he turns to you, the best programmer in the country, to write a program for getting the correct count for the trees in different parts of the jungle.

Add your code only to the parts of the file indicated. Do not modify any other part of the given code, and do not add new files.

Inputs

The first line contains N , the number of trees in the path, and $2 \leq N \leq 500,000$. The next line(s) contains N integers $a_1, a_2, a_3, \dots, a_N$, where a_i is a positive integer that represents the height of the i^{th} tree in the path.

Outputs

An integer, which is the total number of pairs of trees which the monkeys can swing directly from one to the other with the given list of tree heights.

Sample Input 1

```
4
3 4 1 2
```

Sample Output 1

4

Sample Input 2

5

19 17 20 20 20

Sample Output 2

5

Sample Input 3

12

12 8 21 15 9 3 4 6 11 12 18 24

Sample Output 3

20

Additional Requirements

- For every given test case, the output integer is guaranteed to be less than 2^{31} .
- An efficient program is required. Specifically, the time complexity should be $O(N)$. You would get **at most 50% of the marks** if your program's time complexity is worse than $O(N)$.
 - If you check all the pairs to see if there is any tree in between them that will block the swing, it may be too slow to pass some of the test cases.
 - You can improve the efficiency by using a stack.

Submission

You need to submit your completed **Swing.cpp** to CodeCrunch (<https://codecrunch.comp.nus.edu.sg/>) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.