CS1020E | Lab 10 | Exercise 1

Monotonically Increasing Intervals

Objectives

The focus of this exercise is algorithm efficiency and time complexity.

Problem Description

A *monotonically increasing interval* (MII) of an array is a consecutive subsequence of the array in which the elements are monotonically increasing. A MII should have at least two elements. For example, {2, 3, 5} is a monotonically increasing interval of the array {4, 2, 3, 5, 7, 6, 8}.

MIIs are very useful, but you are not going to use them here. Instead, in this exercise, you are to count the number of MIIs of a given array of integers.

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 an integer *N*, which is the number of elements of the array, and 1 <= *N* <= 100,000. The second line contains *N* integers, which are elements of the array.

Outputs

Output the number of MIIs of the input array.

Note: As the number of MIIs for some of our test cases can go beyond 2^{32}, you should use the `long long` type to keep track of the number of MIIs.

Sample Input

```
7
4 2 3 5 7 6 8
```

Sample Output

```
7
```

Explanation

The MIIs are {2, 3}, {2, 3, 5}, {2, 3, 5, 7}, {3, 5}, {3, 5, 7}, {5, 7}, and {6, 8}

Additional Requirement

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).

(Hint: You do not need to find all the MIIs in order to know how many there are.)

Submission

You need to submit your completed **`mii.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.