Problem | Editorial | Submissions | Comments

Roof Top

Difficulty: **Easy** Accuracy: **40.48%** Submissions: **74K+** Points: **2**

You are given the heights of consecutive buildings. You can move from the roof of a building to the roof of the next adjacent building. You need to find the maximum number of consecutive steps you can put forward such that you gain an increase in altitude with each step.

Examples:

Input: arr[] = [1, 2, 2, 3, 2]
Output: 1
Explanation: 1, 2, or 2, 3 are the only consecutive buildings with increasing heights thus maximum number of consecutive steps with an increase in gain in altitude would be 1 in both cases.

Input: arr[] = [1, 2, 3, 4]
Output: 3
Explanation: 1 to 2 to 3 to 4 is the jump of length 3 to have a maximum number of buildings with increasing heights, so maximum steps with increasing altitude becomes 3.

Expected Time Complexity: O(n)
Expected Auxiliary Space: O(1)

Constraints:
$1 <= arr.size() <= 10^6$
$1 <= arr[i] <= 10^5$

Try more examples

Seen this question in a real interview before ? Yes No

Topic Tags

Related Articles

Report An Issue
If you are facing any issue on this page. Please let us know.

Java (1.8) Average Time: 30m Start Timer

```java
//{ Driver Code Starts
import java.io.*;
import java.lang.*;
import java.util.*;

class gfg {

    public static void main(String args[]) throws IOException {
        BufferedReader read = new BufferedReader(new InputStreamReader(Sys
        int t = Integer.parseInt(read.readLine());

        while (t-- > 0) {
            String st[] = read.readLine().trim().split("\\s+");
            int n = st.length;
            int arr[] = new int[n];

            for (int i = 0; i < n; i++) arr[(int)i] = Integer.parseInt(st[(int)

            System.out.println(new Solution().maxStep(arr));
        }
    }
}

// } Driver Code Ends




class Solution {
    // Function to find maximum number of consecutive steps
```

Custom Input Compile & Run Submit