**1. Problem Discussion**

In this problem you will be provided with an array of size ’n’ containing n elements which represents heights of consecutive buildings. 2.You can move from the roof of a building to the roof of the next adjacent building. 3. 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. For Example: Input:

****

Output

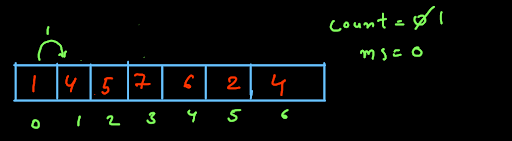
****

**2. Approach**

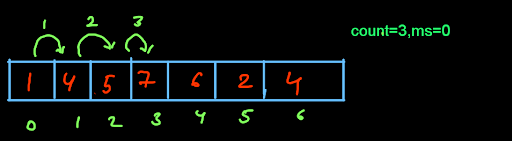
Here we will be solving this problem in O(n) time only by traversing the whole array . We will be working with only variables here .Let's say( ms=0(maximum altitude that we have gained ) and count=0(Number of jump)) (initially).

****

Now we start from the beginning of the array where i=0,For every element we check if the next element is greater than the current element or not . If the arr[i+1] > arr[i] ,it means the next building has more height than the current building so here you have to jump so increase the count to count+1.

****

Then i is increased to i+1 then we again check if arr[i] < arr[i+1] if it is then again increase count to count+1 ,we continue to increase count and i till arr[i]<arr[i+1] once arr[i+1] become lesser than arr[i] we update out ms to count if count > ms otherwise ms contains the previous value and update count to zero.

****

**3. Code**

ConsoleJava

import java.util.\*;

public class Main {

public static int findMaxSteps(int[]arr) {

//write your code here

int ans = 0;

int count = 0;

for(int i=0; i < arr.length-1;i++) {

if(arr[i] < arr[i+1]) {

count++;

}

else {

ans = Math.max(ans,count);

count = 0;

}

}

ans = Math.max(ans,count);

return ans;

}

public static void main(String[]args) {

//input work

Scanner scn = new Scanner(System.in);

int n = scn.nextInt();

int[]arr = new int[n];

for(int i = 0 ; i < n; i++) {

arr[i] = scn.nextInt();

}

int ans = findMaxSteps(arr);

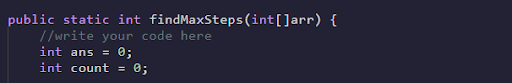
System.out.println(ans);

}

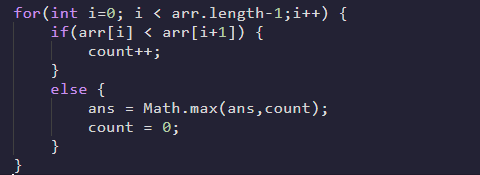
}

**4. Code Explanation**

Here we take two variables ans(Maximum altitude gained) and count (number of jumps)

****

Now we compare each building's height with its next building’s height that is represented by array elements.

****

Here we compare the last number of consecutive jumps with the maximum number of jumps calculated so far. Our ms will be the maximum of count and ms.

****

**5. Analysis**

Time Complexity

O(n) Here the time complexity is O(n) Because we traverse the array only once.

Space Complexity

Constant Here the space complexity is constant as no extra space is used.