HEIGHTS SORTED II

Students are requested to stand in a line for a program.

- The student who came first stood at first and the one who came next stood back at them and every student followed this.The problem is every student is having different heights .Some of them are unable to see whats going in front of them .The Students dont want to change their places
- Now You have a operation to solve the problem so they can able to see the program .
- Operation:
- You can choose one of the student and increment his height by 1cm
- For example:

  ```
  1 1 1 are the heights of the students

  Explanation:
  you choose third person and increased his height then it would be like
  1 1 2
  But student 2 is unable to see whats going on in front
  you choose second person and increased his height then it would be like
  1 2 2
  Still the problem is not solved
  you choose third  person and increased his height then it would be like
  1 2 3


  Note: The student is able to see if the height of the student is minimum
  1cm greater than the before student
  ```

- Find the minimum number of operations to get the problem solved
- Each time the students are called for the program they choose different places ,
- You need to print minimum number of operations each time

Input Format

- First line of input contains **T** no od testcases
- First line of each testcase contains **N** number of students
- Second line of each testcase contains N sepaarate spaced Integers **Ai Ai+1 ...An** Heights of the students

Constraints

- **1 <= T <= 1000**
- **1 <= N <= 5000**
- **1 <= A[i] <=10^4**

Output Format

- For each testcase return minimum number of operations

Sample Input 0

```
3
3
1 1 1
7
```

```
1 2 3 1 4 5 1
9
1 1 3 4 5 6 7 8 1
```

Sample Output 0

```
3
11
9
```

Sample Input 1

```
1
4
10 100 9 1
```

Sample Output 1

```
193
```

EXPLANATION:

Each time you are given heights of the students.Given that the students dont want to change their places .
Usually the first one is able to see no matter what

But from second person the person is only able to see when the height of the person is 1cm greater than the before person

We will make strictly increasing sequence from starting which gives us minimum number of operations

CODE:

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        int t;
        Scanner sc=new Scanner(System.in);
        t=sc.nextInt();
        for(int j=0;j<t;j++){
            int n=sc.nextInt();
            int [] nums=new int [n];
            for(int k=0;k<n;k++){
                nums[k]=sc.nextInt();
            }
           int sum=0;
        for(int i=1;i<nums.length;i++){
          int temp=nums[i];
           nums[i]=Math.max(nums[i-1]+1,nums[i]);
           sum=sum+(nums[i]-temp);
        }
       System.out.println(sum);
        }
    }
}
```

example:
2 1 4 5 3

let array name is nums

the first person of height 2 is able to see the program

Running loop over 1 to N

Checking whether the maximum of (previous height +1,current height)
{nums[i]=max(nums[i-1]+1,nums[i])}assigning the maximum of them to nums[i]

2 1 4 5 3

for i=1
 nums[1]=max(2+1,1)
 nums[1]=3;
 Then it would like 2 3 4 5 3
 The cost is updated value -previous value==> 3 -1=2

To get this store previous value in a temp variable

then i=2

heights=2 3 4 5 3
nums[i]=4
storing temp=nums[i]

nums[i]=max(nums[i-1]+1,nums[i])
              = max(3+1,4)
 nums[i]=4;
 cost=nums[i]-temp =4-4=0;

then i=3
would be same as we need not need to change as the height is already greater

for i=4

heights =2 3 4 5 3

temp=nums[i]

```
nums[i]=max(5+1,3)
nums[i]=6
cost=nums[i]-temp
       =6-3
        =3
```

the total cost is 2+3 =5 {cost =cost+nums[i-1]-temp}will make it possible