Talent Battle 100 Days Coding Series

There are *N* stones in a pond, each having a value *Ai* written on it. A frog is at stone 1 and wants to reach stone *N*. The frog can jump from a stone *i* to any stone *j*(*j*>*i*). Let *d* be the length of subarray (i.e. *j*−*i*+1), then the energy required for the jump is (*d*·*Ai*)−*Aj*. Find the minimum **non-negative** amount of energy required by the frog to reach the *N*-th stone.

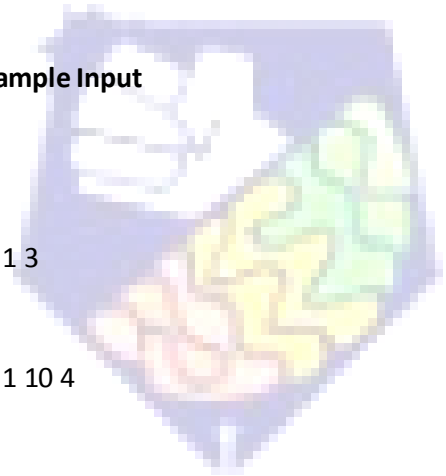
Note: It is possible that the **total** amount of energy required is negative, in that case, you should print the minimum non-negative value (i.e. 0).

Input Format

- The first line contains an integer *T* - the number of test cases. Then the test cases follow.
- The first line of each test case contains an integer *N* - the number of stones.
- The second line contains *N* integers denoting the numbers written on the stones.

Output Format

For each test case output a single integer - the minimum **non-negative** energy required by the frog.



Sample Input

4

3

6 1 3

4

3 1 10 4

3

7 9 1

2

1 5

Sample Output

10

4

20

0

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int32_t main()
{
   int t;
   cin>>t;
   while(t--)
   {
     int n;
     cin>>n;
     int A[n];
     for(int i=0;i<n;i++)
      cin>>A[i];
   int ans=0;
   int a=0;
     for(int i=1;i<n;i++)
     {
        if(A[i]<A[a])
        {
           ans+=(i-a+1)*A[a]-A[i];
           a=i;
        }
     }
     if(a!=n-1)
       ans+=(n-1-a+1)*A[a]-A[n-1];
     if(ans>0)
      cout<<ans<<endl;
```

```
    else
      cout<<0<<endl;
}
}
```

Java

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

class Main
{
   public static void main (String[] args) throws java.lang.Exception
   {
      BufferedReader bu=new BufferedReader(new InputStreamReader(System.in));
      StringBuilder sb=new StringBuilder();
      int t=Integer.parseInt(bu.readLine());
      while(t-->0)
      {
         int n=Integer.parseInt(bu.readLine());
         String s[]=bu.readLine().split(" ");
         int a[]=new int[n],i;
         for(i=0;i<n;i++) a[i]=Integer.parseInt(s[i]);

         long ans=a[0]; int min=a[0];
         for(i=1;i<n;i++)
         {
            ans+=min;
            min=Math.min(min,a[i]);
```

```
            }
            ans-=a[n-1];
            ans=Math.max(ans,0);
            sb.append(ans+"\n");
        }
        System.out.print(sb);
    }
}
```

Python

```
from math import *
for u in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))
    x,y=0,0
    s=0
    while(y<n):
        if(l[y]>=l[x]):
            y=y+1
        else:
            s+=(y-x+1)*l[x]-l[y]
            x=y
    s+=(n-x)*l[x]-l[n-1]
    if(s<=0):
        print(0)
    else:
        print(s)
```