

[解题报告]H Sequence

[Source]

<http://202.114.18.202:8080/judge/contest/view.action?cid=6199#problem/H>

[Description]

对于一个给定的序列 a_1, \dots, a_n ，我们对它进行一个操作 $\text{reduce}(i)$ ，该操作将数列中的元素 a_i 和 a_{i+1} 用一个元素 $\max(a_i, a_{i+1})$ 替代，这样得到一个比原来序列短的新序列。这一操作的代价是 $\max(a_i, a_{i+1})$ 。进行 $n-1$ 次该操作后，可以得到一个长度为 1 的序列。我们的任务是计算代价最小的 reduce 操作步骤，将给定的序列变成长度为 1 的序列。

[Solution]

容易想到一个贪心的解法，从最小的数字开始，考察其两边的数字，与较小的一个合并，总代价加上这个值，然后把这个最小值移除，然后同样处理剩下的数中的最小值。该做法可以用栈实现，如果新的元素比栈顶元素小，继续入栈，如果比栈顶元素大，则合并出栈。

在上个算法中，每一个数字都与两边的数字各比较一次，然后移除自己，并增添两边数值的较小者。也就是说，一个数字只要有一个与其相邻的值比它小，它就会对总代价贡献一次。

于是可以得到一个更简单的做法，只需比较两两相邻的数字，然后加上其他较大的即可。

[Code]

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>

using namespace std;
const int N=1000005;
int a[N];

int main()
{
    int i,n,x,y;
    while (scanf("%d",&n)==1)
```

```
{
    for (i=1; i<=n; i++) scanf("%d",&a[i]);
    long long res=0;
    for (i=2; i<=n; i++)
        res+=max(a[i],a[i-1]);
    cout<<res<<endl;
}
```