2865.美丽塔 I

题目大意

一条直线上共 n 个位置，每个位置都有一个 $maxHeights[i]$ 限制了位置 i 的最大高度，建造一座高度和最大的“山脉”（前半增，后半降）。

题目分析

观察数据范围，$n \leq 1000$，可接受 $O(n^2)$的时间复杂度。$maxHeights[i] \leq 1e9$，数据范围需要开到long long。

首先是一个朴素的想法：直接暴力枚举每一个位置，假装它们是“山顶”，然后贪心地向两边山脚取能取到的最大高度。因为先前位置取的高度越高，后面的限制就越宽松，于是贪心可取得最优解。时间复杂度为 $O(n^2)$，由于从山顶向两侧山脚的计算过程是对称的，现给出向左计算的代码片段：

```
long long thissum = maxHeights[k],nowmax = maxHeights[k];
for(int i=k-1;i>=0;i--){
    nowmax = min((long long)maxHeights[i],nowmax);
    thissum+=nowmax;
}
```

优化

尽管我们无法将大 O 符号内的 n^2 进一步减小，我们可以通过有心地选择一个更有可能的“好山顶”在最外层进行剪枝。观察到一个“山谷”绝不可能是一个“好山顶”，同样地，“山坡”也不会是一个“好山顶”，下面进行简单的证明：

假设存在相邻的两个位置，有 $maxHeights[i] = a$ 与 $maxHeights[i+1] = a + h$，假设在位置 i 的高度为 a 时，位置 0 到 i 高度总和为 $sum1$。当我们选取位置 i 为山顶时，位置 $i+2$ 到 $n-1$ 总和为 $sum2$，从而 $ans1 = sum1 + 2a + sum2$，当我们选取位置 $i+1$ 为山顶时，位置 $i+2$ 到 $n-1$ 总和为 $sum3$，此时位置 i 高度为 a，从而 $ans2 = sum1 + 2a + h + sum3$，显然有 $sum2 \leq sum3$，$h > 0$，从而 $ans1 < ans2$。

因此，只需在循环头部加入如下代码，即可进行一定程度上的剪枝：

```
if(k!=0 and k!=n-1 and !(maxHeights[k] >= maxHeights[k-1]
                                and maxHeights[k] >= maxHeights[k+1])){
```

```
    continue;
}
```

时间复杂度仍为 $O(n^2)$，但是时间复杂度已经大大降低。