



实验舱
青少年编程
走近科学 走进名校

提高算法班

单调栈、单调队列

Mas



单调栈

单调栈 (monotonic stack) 一种特殊的栈

它依然只支持普通的栈的操作,但保证栈内元素的值是单调的

单调栈有以下两个性质

- 从栈顶到栈底的元素是具有单调性
- 越靠近栈顶的元素越后进栈

单调栈主要用于线性时间解决 Next Greater Element 问题

即对序列中每个元素,找到下一个比它大/小的元素

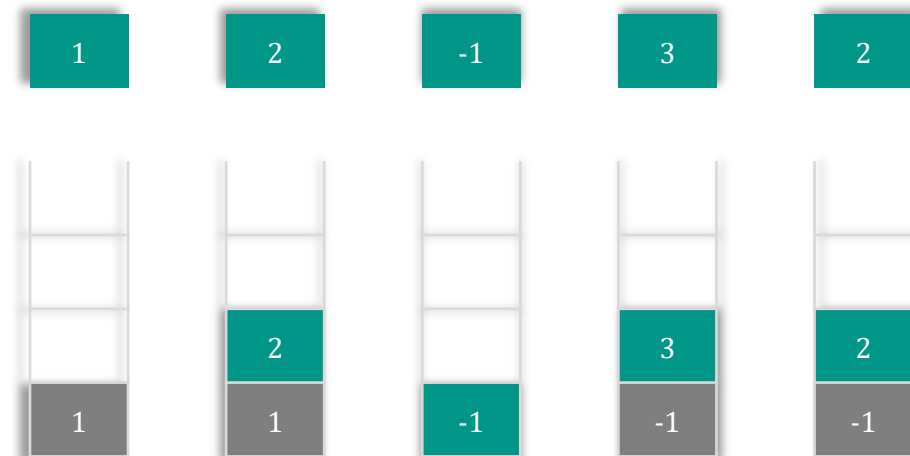
单调栈



实验舱
青少年编程
走近科学 走进名校

维护单调栈,以单调递增栈为例

- 栈为空,当前元素入栈
- 当前元素大于栈顶,入栈
- 弹出栈顶,直到当前元素大于栈顶或栈为空





#2550、单调栈

题目描述

给出项数为 n 的整数数列 $a_1 \dots n$

定义函数 $f(i)$ 代表数列中第 i 个元素之后第一个大于 a_i 的元素的**下标**,即

$$f(i) = \min_{i < j \leq n, a_j > a_i} j$$

若不存在,则 $f(i) = 0$

试求出 $f(1 \sim n)$

输入格式

第一行一个正整数 n 。第二行 n 个正整数 $a_1 \sim n$

输出格式

一行 n 个整数 $f(1 \sim n)$ 的值

数据规模与约定

对于 30% 的数据, $n \leq 100$

对于 60% 的数据, $n \leq 5 \times 10^3$

对于 100% 的数据, $1 \leq n \leq 10^6, 1 \leq a_i \leq 10^9$

输入样例

```
5
1 4 2 3 5
```

输出样例

```
2 5 4 5 0
```



#2550、单调栈

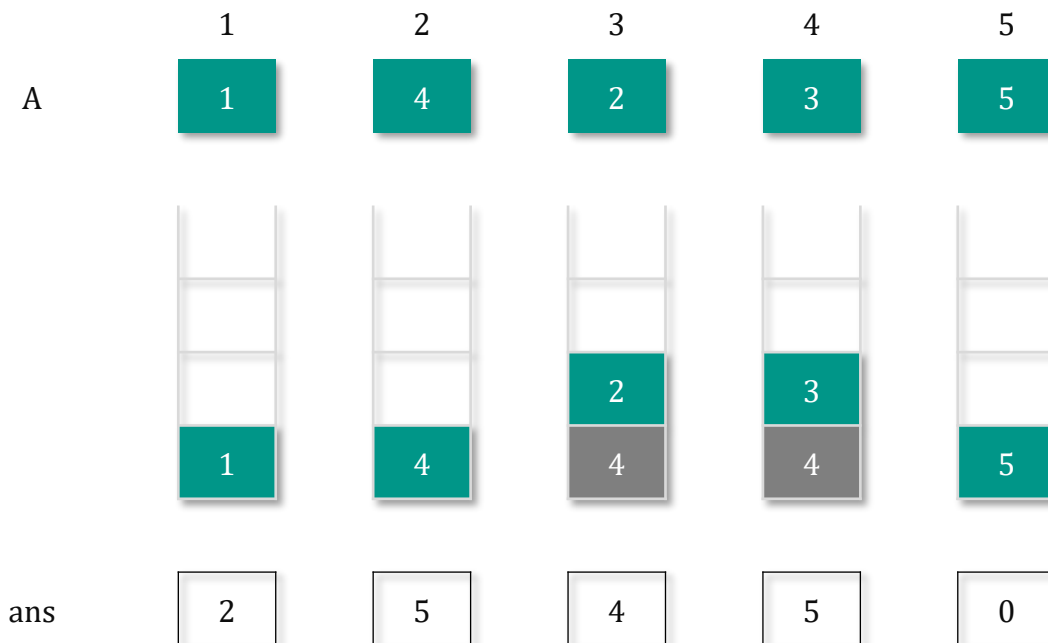
维护单调栈,自栈底到栈顶单调非增

当元素不能直接入栈时

当前元素一定大于需要弹出的元素

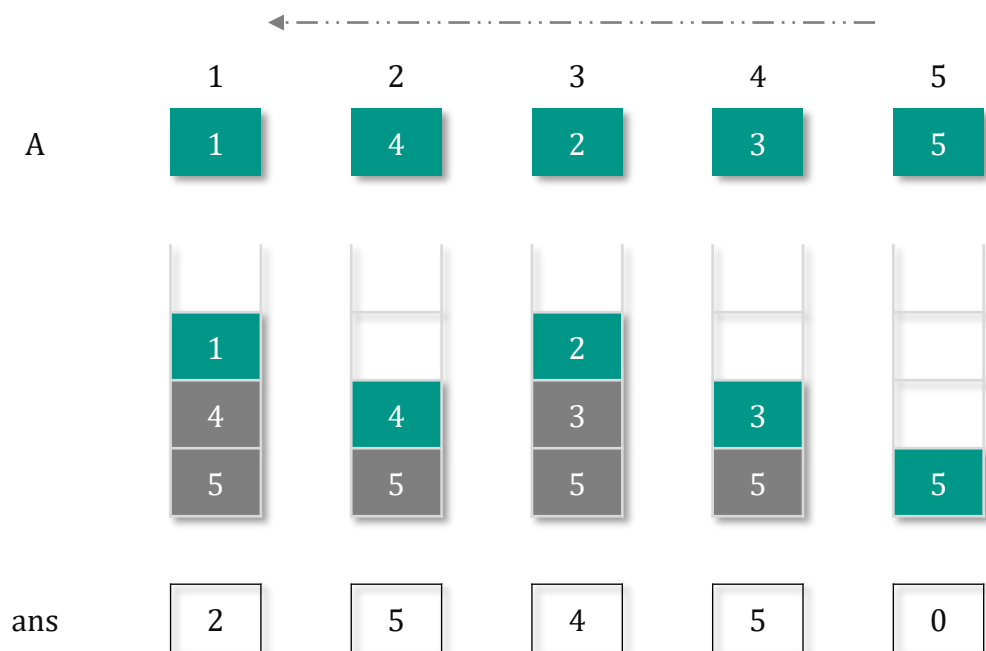
且一定是最接近的,时间复杂度 $O(n)$

某个下标的答案在比它大的数入栈时才被更新





#2550、单调栈



逆序遍历数组,维护一个单调递减栈

当元素可以入栈时

栈顶元素一定大于当前元素

且一定是最近的,时间复杂度 $O(n)$

某个下标的答案在枚举到它时就能得到



#2551、糟糕的一天

题目描述

农夫约翰有 N 头奶牛,每一头牛都站在同一排面朝东方,而且每一头牛的身高为 h_i

第 N 头牛在最前面,而第 1 头牛在最后面

对于第 i 头牛前面的第 j 头牛,如果 $h_i > h_{i+1}$ 并且 $h_{i+1} > \dots > h_j$,那么认为第 i 头牛可以看到第 $i+1 \sim j$ 头牛

定义 C_i 为第 i 头牛所能看到的别的牛的头发的数量。

请帮助农夫约翰求出 $\sum_{i=1}^n C_i$

输入格式

第一行一个正整数 N

接下来 N 行每行一个正整数 h_i

输出格式

输出一个整数,表示 $\sum_{i=1}^n C_i$

输入样例

```
6
10
3
7
4
12
2
```

输出样例

```
5
```

数据规模

对于 40% 的数据 $1 \leq N \leq 10000$

对于全部的数据 $1 \leq N \leq 80000, 1 \leq h_i \leq 10^9$



#2551、糟糕的一天

思路1

每头牛,能看到下一个比它高的牛之间的所有牛

对每头牛求出下一头更高的牛坐标,计算差值累加即可

时间复杂度 $O(n)$

思路2

考虑一头牛能被多少头牛看到

维护单调递减栈

若当前元素比栈顶元素高

那么弹出栈顶(矮的牛会被高的挡住视线),直到栈顶元素大于当前牛的高度

栈内剩余的牛都可以看见这头牛

时间复杂度 $O(n)$

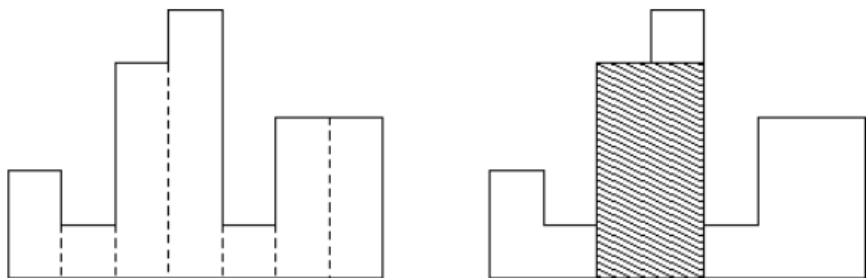


#2547、直方图中最大的矩形

题目描述

直方图是由在公共基线处对齐的一系列矩形组成的多边形,矩形具有相等的宽度,但可以具有不同的高度

例如,图例左侧显示了由高度为 2, 1, 4, 5, 1, 3, 3 的矩形组成的直方图,矩形的宽度都为 1 :



通常直方图用于表示离散分布,例如文本中字符的频率

现在,请你计算在公共基线处对齐的直方图中最大矩形的面积,图例右图显示了所描绘直方图的最大对齐矩形

输入格式

输入包含几个测试用例(不超过 100 组)

每个测试用例占据一行,用以描述一个直方图,并以整数 n 开始,表示组成直方图的矩形数目

然后跟随 n 个整数 $h_1 \sim h_n$

这些数字以从左到右的顺序表示直方图的各个矩形的高度

每个矩形的宽度为 1 ,同行数字用空格隔开

当输入用例为 $n = 0$ 时,结束输入,且该用例不用考虑

输出格式

对于每一个测试用例,输出一个整数,代表指定直方图中最大矩形的区域面积

每个数据占一行,请注意,此矩形必须在公共基线处对齐

输入样例

```
7 2 1 4 5 1 3 3
4 1000 1000 1000 1000
0
```

输出样例

```
8
4000
```

数据范围

对于全部的数据 $1 \leq n \leq 100000, 0 \leq h_i \leq 1000000000$

#2547、直方图中最大的矩形

枚举每一个元素 H_i 作为矩形的最低点

往左找到第一个比他小的元素 H_L ，往右找到第一个比他小的元素 H_R

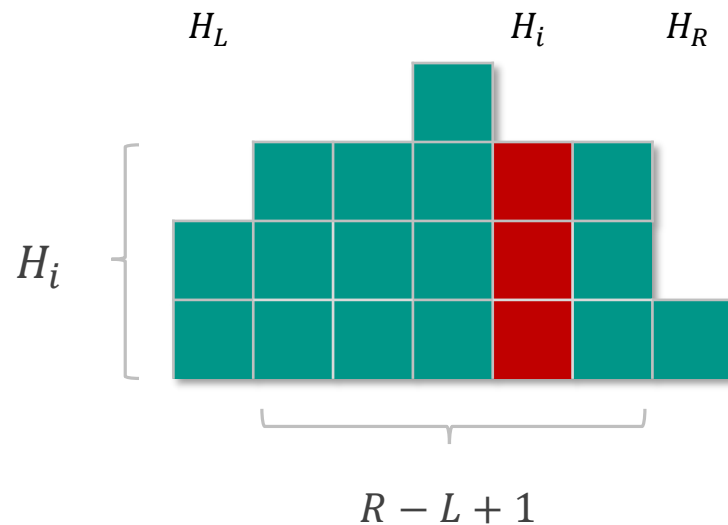
那么 $[L + 1, R - 1]$ 范围的矩形就是合法的

面积为 $(R - L + 1) \times H_i$

时间复杂度 $O(n^2)$

对于求解上/下一个小于它的元素

可以使用单调栈优化,时间复杂度 $O(n)$



#2547、直方图中最大的矩形

维护一个单调递增栈,记录柱子下标,栈内柱子高度递增

若当前元素 H_i 不能直接入栈时 H_i 小于栈顶 H_L

不难发现 $[L + 1, i - 1]$ 范围内所有柱子高度单调非降

若 $[L + 1, i - 1]$ 范围内存在小于 H_L 的高度 H_t

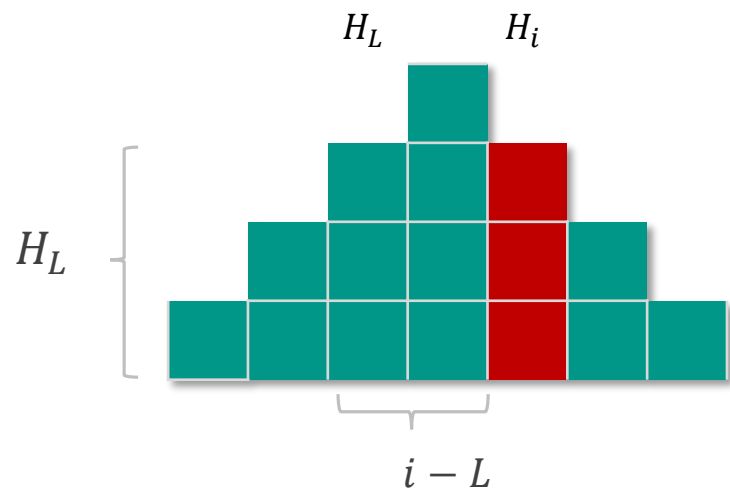
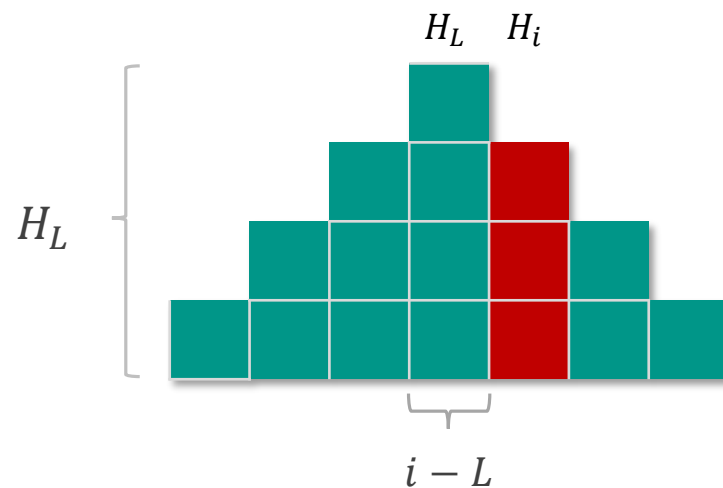
那么 H_t 会将 H_L 弹出栈

即对于由 H_i 引起的弹栈过程中栈内每一个元素可作为矩形高度最低点

将 L 作为左边界 $i - 1$ 为右边界,高度为 H_L

面积为 $H_L \times (i - L)$

是否有遗漏?



#2547、直方图中最大的矩形

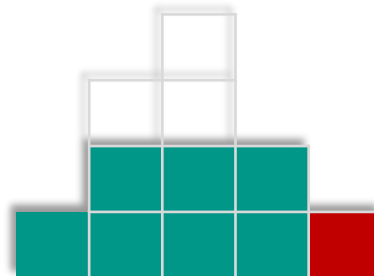
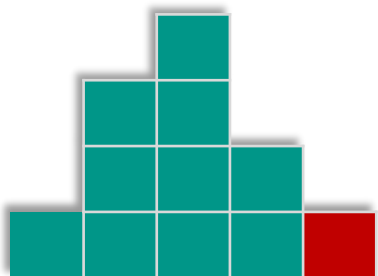
上述计算仅考虑了最低点在最左侧时的情况

对于 H_i 作为最低点且最低点在中间/最右侧时的情况并未考虑

记录最后一个弹栈元素下标 idx ，将其高度修改为 H_i ，将 idx 重新加入栈

该操作相当于将栈内所有 $\geq H_i$ 高度改为 H_i

即将 H_i 作为最低点且在中间/最右侧时的矩形转化为最低点为左侧时矩形



```
stack<long long> s;  
for (int i = 1; i <= n; i++)  
{  
    int idx = i;  
    while (s.size() && a[i] <= a[s.top()])  
    {  
        idx = s.top();  
        long long h = a[idx];  
        ans = max(ans, h * (i - idx));  
        s.pop();  
    }  
    s.push(idx);  
    a[idx] = a[i];  
}
```



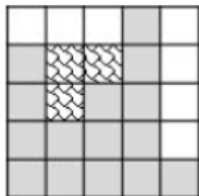
#1974、房屋积水

题目描述

乌龟家的屋顶是凹凸不平的,所以每次雨后都会积水,为了知道屋顶是否会在暴雨后塌掉,他把屋顶的形状给了你,希望你帮他计算暴雨后屋顶的积水总量

乌龟的屋顶由顺次排在同一水平线上的 n 个宽度为 1、高度为整数 (分别给出) 的瓦片组成

例如给定 $n = 5$,瓦片的高度分别为 4, 2, 3, 5, 1,屋顶可以画在下图所示的网格中,灰色格子为瓦片



暴雨过后,如果一个方格向左右两侧延伸都能到达瓦片占据的方格,它就会积水

所以图中波浪线格子在暴雨后会积水,屋顶的积水方格总数为 3

输入格式

两个整数 n, R_1 ,表示屋顶的宽度和生成数列的首项。从左向右数第 $i (1 \leq i \leq n)$ 个瓦片的高度 $a_i = R_i \bmod 10$

试题中使用的生成数列 R 定义如下: 整数 $0 \leq R_1 < 201701$ 在输入中给出。

对于 $i > 1, R_i = (R_{i-1} \times 6807 + 2831) \bmod 201701$

输出格式

一个整数,表示暴雨后屋顶积水方格的总数

输入样例

```
10 1
```

输出样例

```
23
```

数据规模

对于 60% 的数据 $1 \leq n \leq 10^4$

对于 100% 的数据 $1 \leq n \leq 10^6$



#1974、房屋积水

思路1

对于每一个 H_i ，若它能够被左右两个比他高的格子围住，它能够蓄水

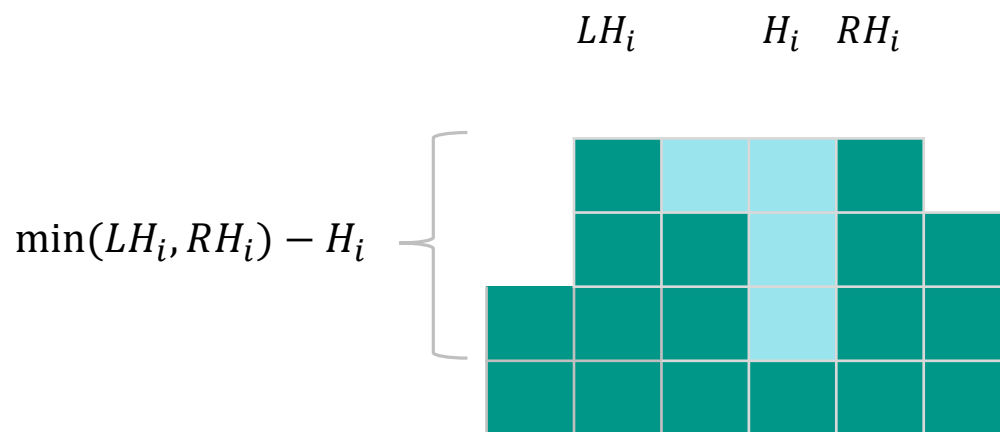
蓄水高度取决于它左/右侧格子最大值的最小值

顺序递推求出 H_i 左边最大高度 LH_i

逆序递推求出右边最大高度 RH_i

累加每列能贡献的蓄水量 $\min(LH_i, RH_i) - H_i$

时间复杂度 $O(n)$





#1974、房屋积水

思路2

维护单调递减栈，当 H_i 不能直接入栈需要弹栈时

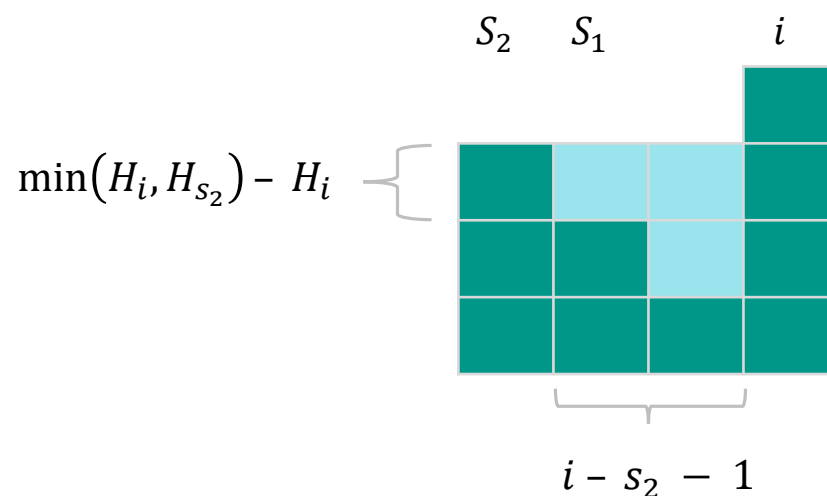
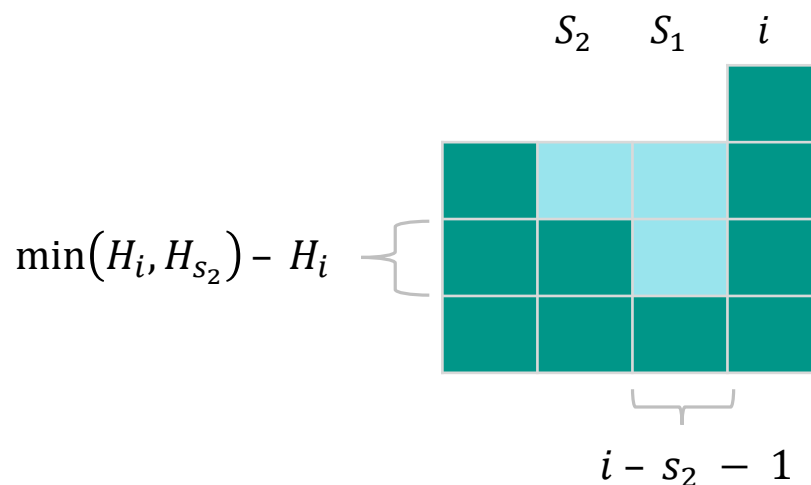
栈顶元素为 H_{s_1} 次顶元素为 H_{s_2} ，且有 $H_{s_2} < H_{s_1} \leq H_i$

此时 H_{s_1} 可以蓄水,贡献蓄水量为

$$(\min(H_i, H_{s_2}) - H_i) \times (i - s_2 - 1)$$

不断弹栈并累加蓄水量即可

时间复杂度 $O(n)$





#2548、玉蟾宫

题目背景

有一天,小猫 *rainbow* 和 *freda* 来到了湘西张家界的天门山玉蟾宫,玉蟾宫宫主蓝兔盛情地款待了它们,并赐予它们一片土地

题目描述

这片土地被分成 $N \times M$ 个格子,每个格子里写着 `R` 或者 `F`, `R` 代表这块土地被赐予了 *rainbow*, `F` 代表这块土地被赐予了 *freda*。现在 *freda* 要在这里卖萌

它要找一块矩形土地,要求这片土地都标着 `F` 并且面积最大

但是 *rainbow* 和 *freda* 的 *OI* 水平都弱爆了,找不出这块土地,而蓝兔也想看 *freda* 卖萌 (她显然是不会编程的.....),所以它们决定,如果你找到的土地面积为 S ,它们每人给你 S 两银子

输入格式

第一行两个整数 N, M ,表示矩形土地有 N 行 M 列

接下来 N 行,每行 M 个用空格隔开的字符 `F` 或 `R`,描述了矩形土地

输出格式

输出一个整数,表示你能得到多少银子,即 最大面积乘 3 的值

说明

对于 50% 的数据, $1 \leq N, M \leq 200$

对于 100% 的数据, $1 \leq N, M \leq 1000$

输入样例

```
5 6
R F F F F F
F F F F F F
R R R F F F
F F F F F F
F F F F F F
```

输出样例

```
45
```




#2548、玉蟾宫

将数组转化为01矩阵

令 $sum[i][j]$ 表示从第 i 行,第 j 列向上延伸的最大长度

可从上到下递推处理

枚举所有行

若忽略行这个维度, $sum[i][j]$ 可看作直方图各柱子高度

转化为 [#2547、直方图中最大的矩形](#), 求直方图中最大矩形面积

时间复杂度 $O(nm)$



单调队列

如果一个选手比你小还比你强,你就可以退役了

——单调队列的原理

单调队列是指一个队列内部的元素具有严格单调性的一种数据结构,分为单调递增队列和单调递减队列

单调队列满足两个性质

- 单调队列必须满足从队头到队尾的严格单调性
- 排在队列前面的比排在队列后面的要先进队

单调队列主要用于解决滑动窗口类问题

即在长度为 n 的序列中,求每个长度为 m 的区间的区间最值



#760、滑动窗口

题目描述

给一个长度为 N 的数组，一个长为 K 的滑动窗体从最左端移至最右端，你只能看到窗口中的 K 个数，每次窗体向右移动一位，如下图：

```
[1 3 -1] -3 5 3 6 7
1 [3 -1 -3] 5 3 6 7
1 3 [-1 -3 5] 3 6 7
1 3 -1 [-3 5 3] 6 7
1 3 -1 -3 [5 3 6] 7
1 3 -1 -3 5 [3 6 7]
```

你的任务是找出窗体在各个位置时的最大值和最小值

输入格式

第 1 行：两个整数 N 和 K

第 2 行： N 个整数，表示数组的 N 个元素

输出格式

第一行为滑动窗口从左向右移动到每个位置时的最小值，每个数之间用一个空格分开 第二行为滑动窗口从左向右移动到每个位置时的最大值，每个数之间用一个空格分开

样例输入

```
8 3
1 3 -1 -3 5 3 6 7
```

样例输出

```
-1 -3 -3 -3 3 3
3 3 5 5 6 7
```

数据范围与提示

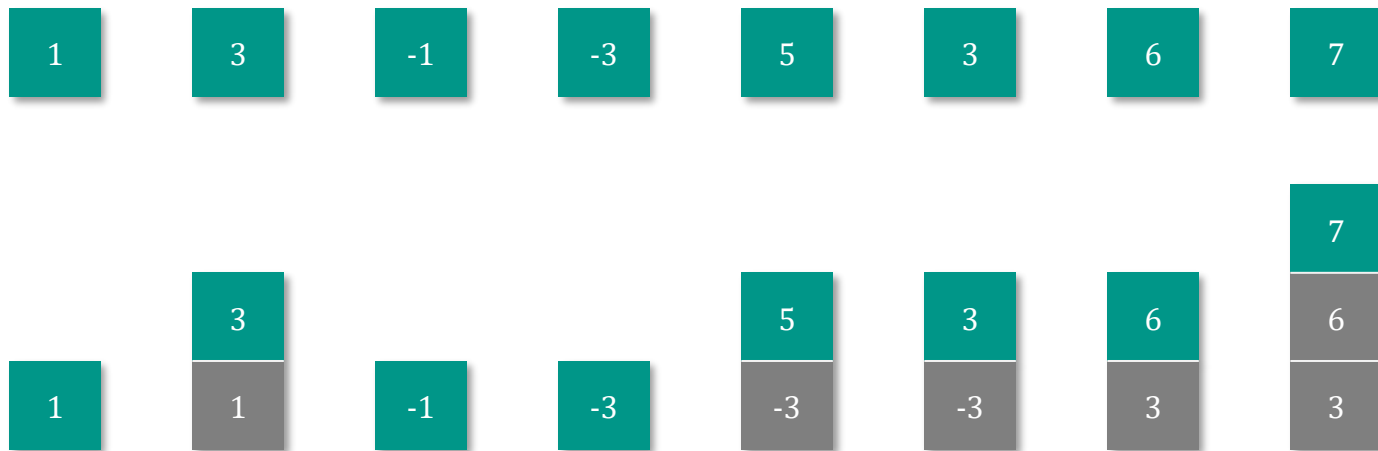
对于 20% 的数据， $K \leq N \leq 1000$ ；

对于 50% 的数据， $K \leq N \leq 10^5$ ；

对于 100% 的数据， $K \leq N \leq 10^6$ 。



#760、滑动窗口



从前往后扫描,以最小值为例

维护一个双端队列(deque)维护单调性,最大的元素一定在队首

如果队首下标与当前元素下标差值超过窗口大小 k ,那么队首出队

如果当前元素比队尾元素大,那么队尾出队



#2555、拥挤的奶牛

题目描述

FJ 有 N 头奶牛沿着一维的栅栏吃草,第 i 头奶牛在目标点 x_i ,它的身高是 h_i .

当一头奶牛左边 D 距离内而且右边 D 距离内有身高至少是它的两倍的奶牛,它就会觉得拥挤

请计算觉得拥挤的奶牛的数量

输入格式

第一行输入两个正整数 n, d

接下来 n 行,每行两个整数 x_i, h_i

输出格式

输出感到拥挤的奶牛数量

数据规模

对于 20% 的数据: $1 \leq N \leq 100$

对于 40% 的数据: $1 \leq N \leq 500$

对于 100% 的数据: $1 \leq N \leq 50000, 1 \leq x_i, h_i, D \leq 10^9$, 保证 x_i 不重复

窗口大小为 D 的 RMQ 问题

维护区间最大值和最小值

统计满足条件的奶牛

时间复杂度 $O(n)$



#761、最大连续和

题目描述

给你一个长度为 n 的整数序列 $\{A_1, A_2, \dots, A_n\}$

要求从中找出一段连续的长度不超过 m 的子序列,使得这个序列的和最大

输入格式

第一行为两个整数 n, m

第二行为 n 个用空格分开的整数序列,每个数的绝对值都小于 1000

输出格式

仅一个整数,表示连续长度不超过 m 的最大子序列和

样例输入

```
6 4
1 -3 5 1 -2 3
```

样例输出

```
7
```

数据范围与提示

对于 50% 的数据, $1 \leq N, M \leq 10^4$

对于 100% 的数据, $1 \leq N, M \leq 2 \times 10^5$



#761、最大连续和

维护前缀和 sum_i

设 $dp[i]$ 为以 A_i 结尾的序列和最大值

$$dp[i] = \max_{\max(0, i-m) \leq j < i} \{ sum_i - sum_j \}$$

朴素枚举时间复杂度 $O(nm)$

不难看出窗口大小为 m

单调队列维护最小值更新答案即可

时间复杂度 $O(n)$



实验舱
青少年编程
走近科学 走进名校

谢谢观看