



单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

# 单调栈

河南省实验中学信息技术组

2026年02月11日



# 单调栈

## 单调栈

河南省实验中学  
信息技术组

## 单调栈

### 例题

乱头发节  
直方图中最大的矩形

### 练习

- 单调栈：栈内元素满足单调性 (递增/递减) 的栈，栈顶的值为栈内元素的最大值 (最小值)。
- 入栈：将一个元素插入单调栈时，为了维护栈的单调性，需要在保证将该元素插入到栈顶后整个栈满足单调性的前提下弹出最少的元素。
- 例如，栈内的元素为 81, 45, 11, 0，需要插入元素 14 时，为了保证单调性需要先弹出元素 0 和 11，然后插入元素 14，栈内元素变为 81, 45, 14。
- 单调栈的应用：及时排除不可能的选项，保持策略集合的高度有效性和有序性。



# 单调栈

## 单调栈

河南省实验中学  
信息技术组

## 单调栈

### 例题

乱头发节  
直方图中最大的矩形

### 练习

```
1 // 单调增
2 stack<int> st;
3 for(int i = 1; i <= n; ++i)
4 {
5     while(!st.empty() && a[i] < st.top())
6         st.pop();
7     st.push(i); // 栈顶为最大值
8 }
```

```
1 // 单调减
2 stack<int> st;
3 for(int i = 1; i <= n; ++i)
4 {
5     while(!st.empty() && a[i] > st.top())
6         st.pop();
7     st.push(i); // 栈顶为最小值
8 }
```



## 【例】乱头发节

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

### 【题目描述】

农夫约翰有  $n$  ( $n \leq 80000$ ) 头奶牛，每头奶牛的身高为  $h_i$  ( $h_i \leq 10^9$ )。这  $n$  头奶牛按照 1 到  $n$  的顺序从左往右站立，所有奶牛都面向右方。

对于奶牛  $i$ ，如果其右侧奶牛的身高低于奶牛  $i$ ，那么奶牛  $i$  就能看到它；若遇到一头奶牛的高度大于等于它的奶牛，那么它就看不到后续的奶牛。请你统计每头奶牛能看到的其他奶牛头数之和。

### 【输入格式】

第一行一个整数  $n$  ( $n \leq 10^5$ )，表示奶牛的头数。

接下来一行  $n$  个整数，其中第  $i$  个整数  $h_i$  ( $h_i \leq 10^9$ ) 表示第  $i$  头奶牛的身高。

### 【输出格式】

一行一个整数，表示所有奶牛能看到的其他奶牛头数之和。

### 【样例输入】

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

### 【样例输出】

```
5
```

### 【样例解释】

这 6 头奶牛能看到的其他奶牛的总头数为  $3 + 0 + 1 + 0 + 1 + 0 = 5$ 。



## 【例】乱头发节

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节

直方图中最大的矩形

练习

- 算法 1
- 某头奶牛若想看到其他奶牛，要求其后的奶牛身高要低于它。
- 定义一个单调递减的栈来存储奶牛：
  - 如果栈为空或者当前奶牛  $i$  的身高低于栈顶奶牛，那么让奶牛入栈。
  - 如果当前奶牛的身高大于等于栈顶奶牛，那么让栈顶一直出栈。出栈的奶牛 (第  $k$  头) 身高必然大于从第  $k+1 \sim i-1$  头奶牛，所以第  $k$  头奶牛能看见的奶牛数为  $i-k-1$ ，累加到答案中即可。所有出栈完成后，让当前奶牛入栈。
  - 最后栈内奶牛的身高单调递减，此时让奶牛依次出栈，这些奶牛对答案的贡献为  $n-k$ 。也可以增加一头高度为  $10^9$  的奶牛，这样可以将栈内所有奶牛出栈计算。



## 【例】乱头发节

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

```
1 h[n + 1] = 1e9 + 10; // 增加一头奶牛
2 long long ans = 0;
3 stack<int> st;
4 for(int i = 1; i <= n + 1; ++i)
5 {
6     while(!st.empty() && h[i] >= h[st.top()])
7     {
8         ans += i - st.top() - 1; // 例如 8 7 6 5 10
9         st.pop();
10    }
11    st.push(i);
12 }
```



## 【例】乱头发节

### 单调栈

河南省实验中学  
信息技术组

### 单调栈

### 例题

乱头发节  
直方图中最大的矩形

### 练习

- 算法 2
- 奶牛  $i$  能看到的奶牛为紧跟其后的连续的身高低于它的奶牛，如果遇到身高大于等于它的，则看不到后续的奶牛。
- 定义一个单调递减的栈来存储奶牛：
  - 如果栈为空或者当前奶牛  $i$  的身高低于栈顶奶牛，那么让奶牛入栈。
  - 如果当前奶牛的身高大于等于栈顶奶牛，那么让栈顶一直出栈。所有出栈完成后，让当前奶牛入栈。
  - 最后栈内奶牛的身高单调递减，此时让奶牛依次出栈。也可以增加一头高度为  $10^9$  的奶牛，这样可以将栈内所有奶牛出栈。
  - 奶牛入栈时，栈为空或者栈内的奶牛身高均高于它，所以这头奶牛会被栈内所有的奶牛看见。



## 【例】乱头发节

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

```
1 h[n + 1] = 1e9 + 10; // 增加一头奶牛
2 long long ans = 0;
3 stack<int> st;
4 for(int i = 1; i <= n + 1; ++i)
5 {
6     while(!st.empty() && h[i] >= h[st.top()])
7         st.pop();
8     ans += st.size(); // 第 i 头奶牛被多少奶牛看见
9     st.push(i);
10 }
11 cout << ans;
```



## 【例】乱头发节

### 单调栈

河南省实验中学  
信息技术组

### 单调栈

### 例题

乱头发节  
直方图中最大的矩形

### 练习

- 算法 3
- 奶牛  $i$  能看到的奶牛为紧跟其后的连续的身高低于它的奶牛，如果遇到身高大于等于它的，则看不到后续的奶牛。
- 定义  $cnt[i]$  表示第  $i$  头奶牛能看到的奶牛数。
- 定义单调递减的栈来存储奶牛：
  - 如果栈为空或者当前奶牛的身高低于栈顶奶牛，那么让奶牛入栈。
  - 如果当前奶牛的身高大于等于栈顶奶牛，那么让栈顶一直出栈。出栈的奶牛(第  $k$  头) 身高必然小于出栈后栈顶的奶牛(栈非空)，而且出栈的奶牛能看到的奶牛必然能被栈顶奶牛看见，于是记录栈顶奶牛能够看到的奶牛头数加上  $1 + cnt[k]$ 。所有出栈完成后，让当前奶牛入栈。
  - 最后，栈内奶牛的身高单调递减，按照上述统计方法让栈内所有奶牛出栈。或者增加一头高度为  $10^9$  的奶牛，就可以无需清空栈。



## 【例】乱头发节

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

```
1 h[n + 1] = 1e9 + 10; // 增加一头奶牛
2 stack<int> st;
3 for(int i = 1; i <= n + 1; ++i)
4 {
5     while(!st.empty() && h[i] >= h[st.top()])
6     {
7         int k = st.top(); st.pop();
8         // 栈顶能看见奶牛 k 及奶牛 k 能看见的所有奶牛
9         if(!st.empty()) cnt[st.top()] += 1 + cnt[k];
10    }
11    st.push(i);
12 }
13 long long ans = 0;
14 for(int i = 1; i <= n; ++i) ans += cnt[i];
```



## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

单调栈

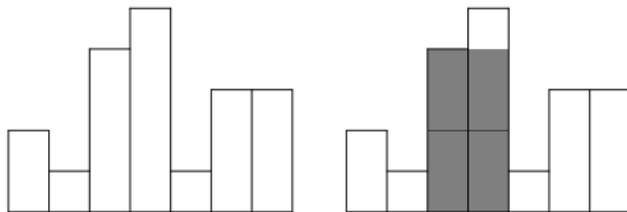
例题

乱头发节  
直方图中最大的矩形

练习

### 【题目描述】

在一条水平线上方有  $n$  个宽为 1 的矩形形，求包含于这些矩形的并集内部的最大矩形的面积 (如下图中的阴影部分面积)。



### 【输入格式】

第一行一个整数  $n$  ( $n \leq 10^5$ ), 表示矩形的个数。

接下来一行  $n$  个整数, 其中第  $i$  个整数  $h_i$  ( $h_i \leq 10^9$ ) 表示第  $i$  个矩形的高度。

### 【输出格式】

一行一个整数, 表示最大的面积。

### 【样例输入】

```
7
2 1 4 5 1 3 3
```

### 【样例输出】

```
8
```



## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

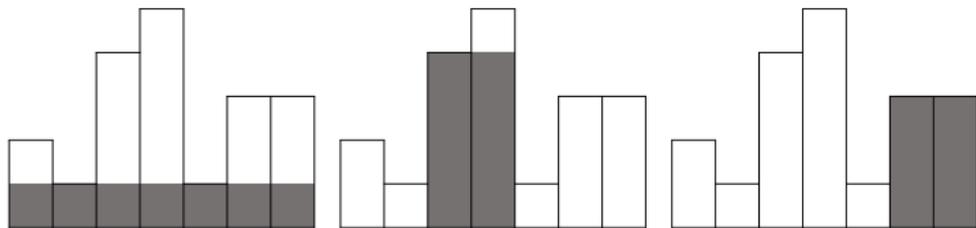
单调栈

例题

乱头发节  
直方图中最大的矩形

练习

- 算法 1
- 可以观察到，构成的大矩形的高度必然等于某一个矩形的高度。



- 枚举以矩形  $i$  的高度构成最大矩形，则需要知道其向左能延申到哪个矩形 (设其为  $l[i]$ )，以及其向右能延申到哪个矩形 (设其为  $r[i]$ )。那么，答案显然为

$$\max_{1 \leq i \leq n} h[i] \times (r[i] - l[i] + 1)$$

- 如何求  $l[i]$  和  $r[i]$ ?



## 【例】直方图中最大的矩形

### 单调栈

河南省实验中学  
信息技术组

### 单调栈

### 例题

乱头度节  
直方图中最大的矩形

### 练习

- 以求  $l[i]$  为例，定义一个矩形高度单调递增的栈。
- 对于当前矩形  $i$ ，若栈顶矩形高度大于等于  $h[i]$ ，那么需要将栈顶的矩形出栈直到栈为空或者栈顶矩形的高度小于  $h[i]$ 。
- 那么，从栈顶 (栈非空时) 到当前矩形  $i$  中间的所有矩形的高度必然都大于等于  $h[i]$ 。
- 如果栈空，说明从第 1 个矩形开始所有的矩形高度都大于等于  $h[i]$ 。
- 当所有高度大于等于  $h[i]$  的矩形出栈后，令当前矩形入栈。



## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

```
1 stack<int> st;
2 for(int i = 1; i <= n; ++i)
3 {
4     while(!st.empty() && h[i] <= h[st.top()])
5         st.pop();
6     if(st.empty()) l[i] = 1;
7     else l[i] = st.top() + 1;
8     st.push(i);
9 }
10 while(!st.empty()) st.pop();
11 for(int i = n; i >= 1; --i)
12 {
13     while(!st.empty() && h[i] <= h[st.top()])
14         st.pop();
15     if(st.empty()) r[i] = n;
16     else r[i] = st.top() - 1;
17     st.push(i);
18 }
19 long long ans = 0;
20 for(int i = 1; i <= n; ++i)
21     ans = max(ans, (long long)h[i] * (r[i] - l[i] + 1));
```



## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

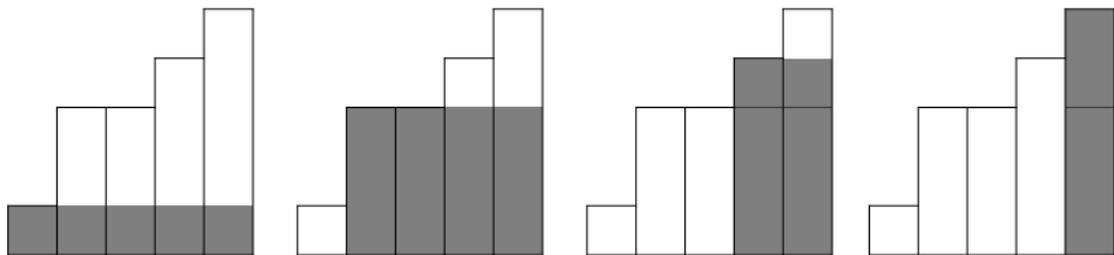
单调栈

例题

乱头度节  
直方图中最大的矩形

练习

- 算法 2
- 首先思考这样一个问题：如果矩形的高度从左到右单调递增，那么答案是多少？
- 显然可以尝试以每个矩形的高度作为最终矩形的高度，并把宽度延申到右边界得到一个矩形，在所有的这样的矩形中取一个面积最大的就是答案。





## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

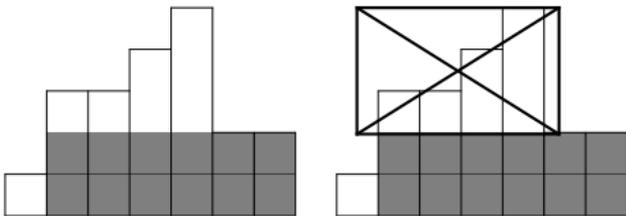
单调栈

例题

乱头发节  
直方图中最大的矩形

练习

- 如果下一个矩形的高度比上一个小，那么要以该矩形的高度来构成大矩形，则如下图中打叉的那部分形状就没有任何用处。



- 既然没有用处，那么不妨将比该矩形高的矩形都删掉，用一个宽度累加、高度为该矩形高度的新矩形来代替被删掉的矩形，这样并不会对后续的计算结果产生影响，于是问题变成了一个矩形高度单调递增的问题。



## 【例】直方图中最大的矩形

### 单调栈

河南省实验中学  
信息技术组

### 单调栈

### 例题

乱头疯节  
直方图中最大的矩形

### 练习

- 建立一个栈，用来保存矩形，要求栈内矩形的高度单调递增(单调栈)。
- 从左往右依次扫描每个矩形：
  - 如果当前矩形比栈顶矩形高，直接进栈。
  - 否则不断取出栈顶，直至栈顶为空或者栈顶矩形的高度比当前矩形小。在出栈过程中，我们累计被弹出的矩形宽度之和，并且每弹出一个矩形，就用它的高度乘上累计的宽度取更新答案即可。整个出栈过程结束后，把一个高度为当前矩形高度、宽度为累计值的新矩形入栈。
- 整个扫描结束后，把栈内剩余的矩形依次弹出，按照上述递增的算法更新答案。也可以增加一个高度为 0 的矩形，最后不用清理栈。



## 【例】直方图中最大的矩形

单调栈

河南省实验中学  
信息技术组

单调栈

例题

乱头发节  
直方图中最大的矩形

练习

```
1 a[n + 1].h = 0, a[n + 1].w = 1; // 最右侧增加一个高度为 0 的矩形
2 long long ans = 0;
3 stack<Node> st;
4 for(int i = 1; i <= n + 1; ++i)
5 {
6     if(st.empty() || a[i].h > st.top().h) st.push(a[i]); // 高度递增
7     else // 高度下降
8     {
9         int w = 0;
10        while(!st.empty() && a[i].h <= st.top().h)
11        {
12            w += st.top().w;
13            ans = max(ans, (long long)w * st.top().h);
14            st.pop();
15        }
16        st.push({a[i].h, w + 1});
17    }
18 }
```



# 练习

## 单调栈

河南省实验中学  
信息技术组

## 单调栈

### 例题

乱头发节  
直方图中最大的矩形

## 练习

- Bad Hair Day(COGS 1746)
- 直方图中最大的矩形 (COGS 3454)
- 玉蟾宫/City Game(COGS 1237)
- 剪切矩形 (COGS 504)
- PLA-Postering(洛谷 P3467)