

{% note info %}

摘要

Title: 131. 直方图中最大的矩形

Tag: 单调栈

Memory Limit: 64 MB

Time Limit: 1000 ms

{% endnote %}

Powered by: NEFU AB-IN

[Link](#)

[@TOC](#)

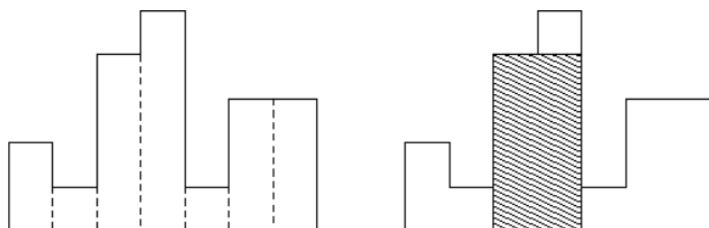
131. 直方图中最大的矩形

• 题意

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。

• 思路

单调栈典型的应用题

具体思路为，枚举每个高度，找到此高度往左和往右延伸的最长长度，两者相乘就是答案

优化思路就是用单调栈实现的，此高度延伸的边界就是比它小的高度，那么就是要找左边第一个小的 l ，和右边第一个小的 r ， (l,r) 便是此高度能延伸的

注意

- 为了方便处理边界，两侧边界都为-1，即比最小的高度还小
- 此次栈存的是下标，所以栈的初始值是0或者n+1

• 代码

```
'''
Author: NEFU AB-IN
Date: 2022-02-27 09:19:29
FilePath: \ACM\Acwing\131.py
LastEditTime: 2022-02-27 09:52:36
'''

N = int(1e5 + 100)
h, l, r = [0] * N, [0] * N, [0] * N

while True:
    try:
        h = list(map(int, input().split()))[1:]
        if not h:
            break
        n = len(h)
        h = [-1, *h, -1]
        # 此次栈存的是下标，比较时换成下标比较即可
        stk = [0]
        for i in range(1, n + 1):
            while stk and h[stk[-1]] >= h[i]:
                stk.pop()
            l[i] = stk[-1] if stk else -1
            stk.append(i)

        stk = [n + 1]

        for i in range(n, -1, -1):
            while stk and h[stk[-1]] >= h[i]:
                stk.pop()
            r[i] = stk[-1] if stk else -1
            stk.append(i)

        res = 0
        for i in range(1, n + 1):
            res = max(res, h[i] * (r[i] - l[i] - 1))
        print(res)
    except:
```

break