

0084. 柱状图中最大的矩形

👤 ITCharge ⌚ 大约 2 分钟

- 标签：栈、数组、单调栈
- 难度：困难

题目链接

- [0084. 柱状图中最大的矩形 - 力扣](#)

题目大意

给定一个非负整数数组 `heights`，`heights[i]` 用来表示柱状图中各个柱子的高度。每个柱子彼此相邻，且宽度为 1。

要求：计算出在该柱状图中，能够勾勒出来的矩形的最大面积。

解题思路

思路一：枚举「宽度」。一重循环枚举所有柱子，第二重循环遍历柱子右侧的柱子，所得的宽度就是两根柱子形成区间的宽度，高度就是这段区间中的最小高度。然后计算出对应面积，记录并更新最大面积。这样下来，时间复杂度为 $O(n^2)$ 。

思路二：枚举「高度」。一重循环枚举所有柱子，以柱子高度为当前矩形高度，然后向两侧延伸，遇到小于当前矩形高度的情况就停止。然后计算当前矩形面积，记录并更新最大面积。这样下来，时间复杂度也是 $O(n^2)$ 。

思路三：利用「单调栈」减少两侧延伸的复杂度。

- 枚举所有柱子。
- 如果当前柱子高度较大，大于等于栈顶柱体的高度，则直接将当前柱体入栈。
- 如果当前柱体高度较小，小于栈顶柱体的高度，则一直出栈，直到当前柱体大于等于栈顶柱体高度。
 - 出栈后，说明当前柱体是出栈柱体向右找到的第一个小于当前柱体高度的柱体，那么就可以向右将宽度扩展到当前柱体。
 - 出栈后，说明新的栈顶柱体是出栈柱体向左找到的第一个小于新的栈顶柱体高度的柱体，那么就可以向左将宽度扩展到新的栈顶柱体。

- 以新的栈顶柱体为左边界，当前柱体为右边界，以出栈柱体为高度。计算矩形面积，然后记录并更新最大面积。

代码

```
class Solution:
    def largestRectangleArea(self, heights: List[int]) -> int:
        heights.append(0)
        ans = 0
        stack = []
        for i in range(len(heights)):
            while stack and heights[stack[-1]] >= heights[i]:
                cur = stack.pop(-1)
                left = stack[-1] + 1 if stack else 0
                right = i - 1
                ans = max(ans, (right - left + 1) * heights[cur])
            stack.append(i)

        return ans
```

py