

单调栈

Leetcode 84柱状图中的最大矩形

1. 找到每个柱形条左边和右边最近的比自己低的矩形条，然后用宽度乘上当前柱形条的高度作为备选答案。
2. 此类问题的经典做法是单调栈，维护一个单调递增的栈，如果当前柱形条 i 的高度比栈顶要低，则栈顶元素 cur 出栈。出栈后， cur 右边第一个比它低的柱形条就是 i ，左边第一个比它低的柱形条是当前栈中的 top 。不断出栈直到栈为空或者柱形条 i 不再比 top 低。
3. 每个 i 都入栈
4. 最后append一个高度为-1的柱形条让所有柱子出栈

```
class Solution:
    def largestRectangleArea(self, heights: List[int]) -> int:
        stk = []
        heights.append(-1)
        n = len(heights)
        res = 0
        for i in range(n):
            while stk and heights[stk[-1]] > heights[i]:
                cur = stk.pop()
                #如果栈中没有元素 说明cur左边没有比cur更小的了,则最左可以到0(最大宽度
                #所以当前最大矩形面积=
                res = max(res, (i - stk[-1] - 1) * heights[cur])
                #否则 当前高度的宽度=i(右边比cur小的有右边界)-stk[-1](左边比cur小的
                #左边界)-1
            else:
                res = max(res, i * heights[cur])
            stk.append(i)
        return res
```

Leetcode 85最大矩形

```
1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
----- 比如到第三行 heights = [3,1,3,2,2]
1 0 0 1 0
```

则问题转化为84题

```
| |
| | | |
```

```
| | | | |
```

逐行递归 对每一行

得到每行中每一列的高度`heights[i][j]`

1. 如果`mat[i][j]==0`则对当前行"柱子"高度`heights[j]=0`
2. 否则`heights[i][j]=heights[i-1][j]+1`

为了省空间 因为前一行`heights[j]`用完后对于下一行就是`heights[i-1][j]+1`,则`heights[i-1][j]+1`完全可以用一行来更新:

即`heights[j] = heights[j]+1`

```
class Solution:
    def maximalRectangle(self, matrix: List[List[str]]) -> int:
        if not matrix:
            return 0
        m,n = len(matrix),len(matrix[0])
        height = [0]*(n+1)
        res = 0
        for i in range(m):
            stk = []
            for j in range(n+1):
                #height[n]=0是哨兵 用来把所有柱子弄出来的 所以不用设值也不能设值
                if j<n:
                    if matrix[i][j]=='1':
                        height[j]+=1
                    else:
                        height[j]=0
                while stk and height[j]<height[stk[-1]]:
                    cur = stk.pop()
                    if stk:
                        res = max(res,(j-stk[-1]-1)*height[cur])
                    else:
                        res = max(res,j*height[cur])
                stk.append(j)
            return res
```