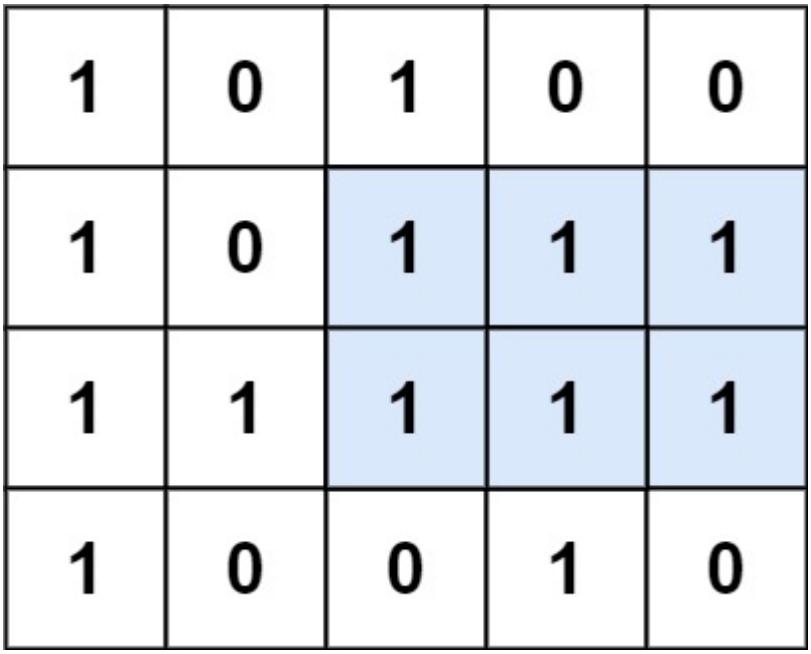
85. Maximal Rectangle

Given a `rows x cols` binary `matrix` filled with `0`'s and `1`'s, find the largest rectangle containing only `1`'s and return *its area*.

Example 1:




```
Input: matrix = [["1","0","1","0","0"],["1","0","1","1","1"],
["1","1","1","1","1"],["1","0","0","1","0"]]
Output: 6
Explanation: The maximal rectangle is shown in the above picture.
```

Example 2:

```
Input: matrix = []
Output: 0
```

Example 3:

```
Input: matrix = [["0"]]
Output: 0
```

Example 4:

```
Input: matrix = [["1"]]
Output: 1
```

Example 5:

```
Input: matrix = [["0","0"]]
Output: 0
```

Constraints:

- `rows == matrix.length`
- `cols == matrix[i].length`
- `0 <= row, cols <= 200`
- `matrix[i][j]` is `'0'` or `'1'`.

```python
class Solution:
    def maximalRectangle(self, matrix: List[List[str]]) -> int:
        if len(matrix)==0:
            return 0

        height = [0]*len(matrix[0])
        maxArea = -1
        for i in range(len(matrix)):
            for j in range(len(matrix[0])):
                if matrix[i][j]=='0':
                    height[j]= 0
                else:
                    height[j] = height[j] + 1
            # print(height)
            temp = self.maxAreaHistogram(height)
            maxArea = max(temp,maxArea)
        return maxArea

    def maxAreaHistogram(self,heights):
        n = len(heights)
        smallestRight = [None]*n
        smallestLeft = [None]*n
        stack = []
        for i in range(n-1,-1,-1):
            while len(stack)>0 and heights[i]<=heights[stack[-1]]:
                stack.pop()
            smallestRight[i] = stack[-1] if len(stack) else n
            stack.append(i)
        stack = []

        for i in range(n):
            while len(stack)>0 and heights[i]<=heights[stack[-1]]:
                stack.pop()
```

```
            smallestLeft[i] = stack[-1] if len(stack) else -1
            stack.append(i)

        # print(smallestRight)
        # print(smallestLeft)
        maxArea = 0
        for i in range(n):
            width = abs(smallestRight[i]-smallestLeft[i])-1
            area = width*heights[i]
            maxArea = max(maxArea,area)
        return maxArea
```