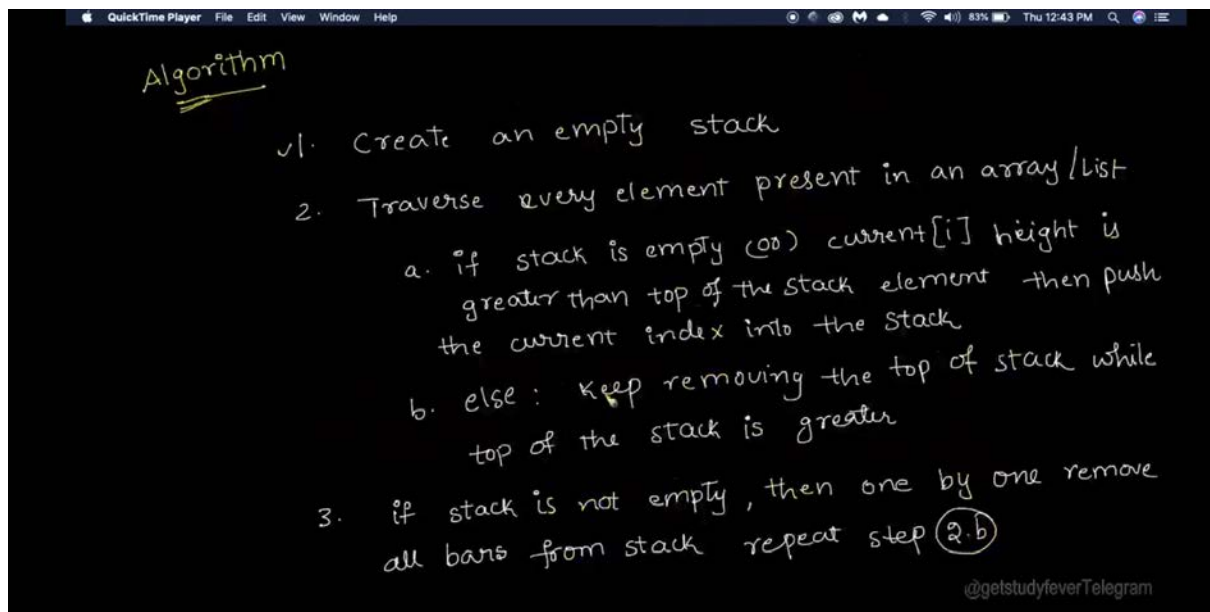


<https://leetcode.com/problems/largest-rectangle-in-histogram/description/>



Time : $O(n)$, space: $O(n)$

```

class Solution:
    def largestRectangleArea(self, heights: List[int]) -> int:
        stack = []
        maxarea = -1

        for i in range(len(heights)):
            if not stack or heights[i] >= heights[stack[-1]]:
                stack.append(i)
            else:
                while stack and heights[stack[-1]] > heights[i]:
                    index = stack.pop()
                    area = -1
                    if not stack:
                        area = heights[index]*i
                    else:
                        area = heights[index]*(i-stack[-1]-1)
                    if area > maxarea:
                        print(index,area)
                        maxarea = area
                stack.append(i)

        i = len(heights)

        while stack:
            index = stack.pop()
            area = -1
            if not stack:
                area = heights[index]*i
            else:
                area = heights[index]*(i-stack[-1]-1)
            if area > maxarea:
                print(index,area)
                maxarea = area

        return maxarea

```

Applied teacher code

```

class Solution(object):
    def largestRectangleArea(self, heights):
        """
        :type heights: List[int]
        :rtype: int
        """
        stack = []
        maxArea = 0
        index = 0
        while index < len(heights):
            if (not stack) or (heights[index] >= heights[stack[-1]]):
                stack.append(index)
            else:
                topOfStack = stack.pop()
                currentArea = heights[topOfStack] * ((index - stack[-1] - 1) if stack else index)
                maxArea = max(currentArea, maxArea)
        while stack:
            topOfStack = stack.pop()
            currentArea = heights[topOfStack] * ((index - stack[-1] - 1) if stack else index)
            maxArea = max(currentArea, maxArea)
        return maxArea

```

@getstudyfever Telegram

Approach

BETTER CODE:

<https://leetcode.com/problems/largest-rectangle-in-histogram/solutions/28917/ac-python-clean-solution-using-stack-76ms/>

Brute force: Try to find out rectangles of all the height

eg: [2,1,5,6,2,3] height width

Rectangle of height 2: $2 * 1$

Rectangle of height 1: $1 * 6$

Rectangle of height 5: $5 * 2$

Rectangle of height 6: $6 * 1$

Rectangle of height 2: $2 * 4$

Rectangle of height 3: $3 * 1$

Approach: For each height, i will try to find out its max width by expanding its left index and right index

Time: $O(n^2)$ **space:** $O(1)$

```
class Solution:
    def largestRectangleArea(self, heights: List[int]) -> int:
        maxheight = -sys.maxsize

        for i in range(len(heights)):
            leftindex = i
            rightindex = i
            while leftindex >= 0 and heights[i] <= heights[leftindex]:
                leftindex -= 1

            leftindex += 1

            while rightindex < len(heights) and heights[i] <= heights[rightindex]:
                rightindex += 1

            rightindex -= 1

            if (rightindex - leftindex + 1) * heights[i] > maxheight:
                maxheight = (rightindex - leftindex + 1) * heights[i]
        return maxheight
```
