Container With Most Water

Question:

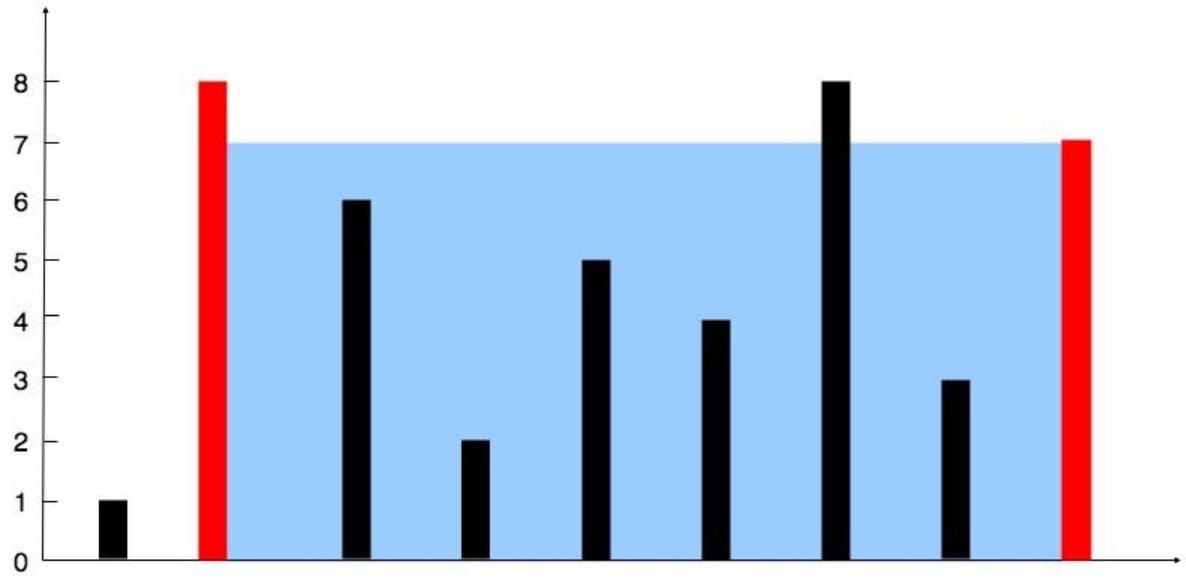
https://leetcode.com/problems/container-with-most-water/

You are given an integer array `height` of length `n`. There are `n` vertical lines drawn such that the two endpoints of the `ith` line are `(i, 0)` and `(i, height[i])`.

Find two lines that together with the x-axis form a container, such that the container contains the most water.

Return the maximum amount of water a container can store.

Example 1:



Input: height = [1,8,6,2,5,4,8,3,7]
Output: 49
Explanation: The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

Example 2:

Input: height = [1,1]
Output: 1

Approach:

The aim here is to find the minimum height from both left and right and find the area between the left and right range.
The left and right show that we need two pointers. One from the left and the other from the right.
Find the minimum between the two heights and calculate the area.
If current area > previous max area, then update the area.
If the minimum height was from the left, increase the left pointer otherwise decrease the right pointer.

Solution:

```
def maxArea(self, height: List[int]) -> int:
        l,r=0,len(height)-1
        c=0
        while(l<r):
                h=min(height[l],height[r])
                c=max(c,h*(r-l))
                l+=(height[l]==h)
                r-=(height[r]==h)
        return c
```

Time Complexity: O(n)

Space Complexity: O(1)