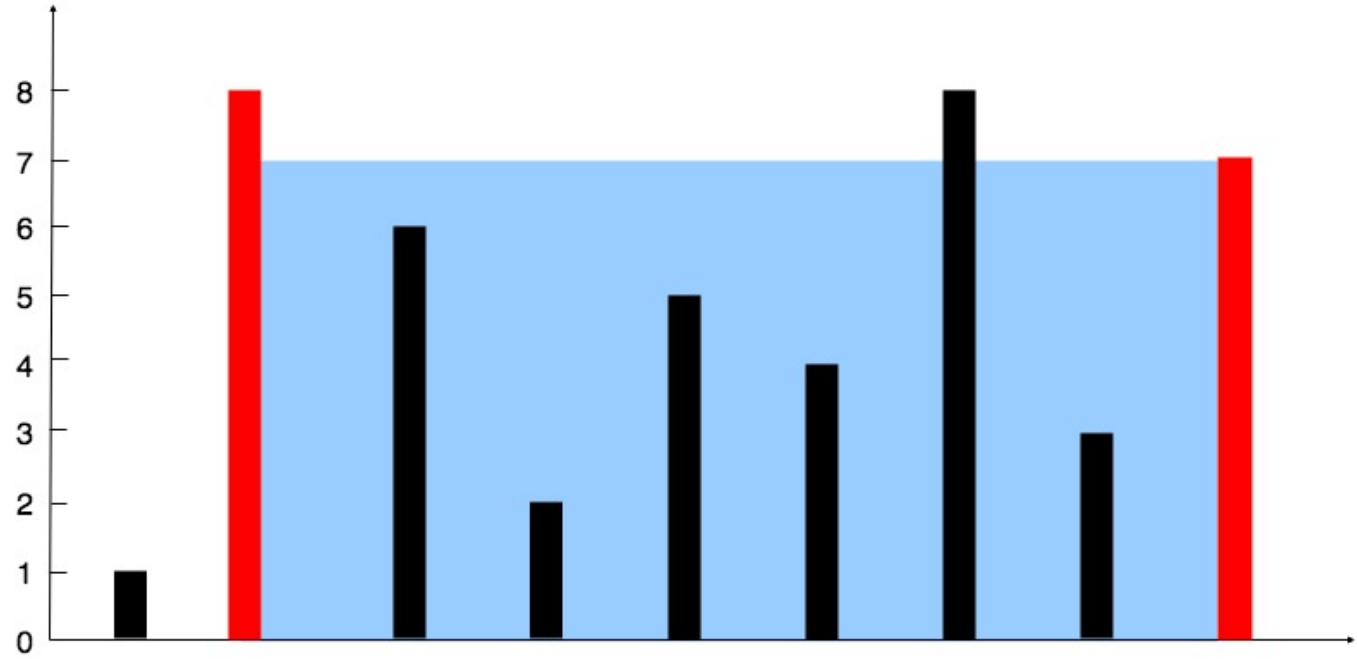
11. Container With Most Water

Given `n` non-negative integers `a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>` , where each represents a point at coordinate `(i, a<sub>i</sub>)`. `n` vertical lines are drawn such that the two endpoints of the line `i` is at `(i, a<sub>i</sub>)` and `(i, 0)`. Find two lines, which, together with the x-axis forms a container, such that the container contains the most water.

Notice that you may not slant the container.

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
```

Example 3:

```
Input: height = [4,3,2,1,4]
Output: 16
```

Example 4:

```
Input: height = [1,2,1]
Output: 2
```

Constraints:

- `n == height.length`
- `2 <= n <= 10<sup>5</sup>`
- `0 <= height[i] <= 10<sup>4</sup>`
- ```
  class Solution:
      def maxArea(self, height: List[int]) -> int:
          maxarea = 0
          i = 0
          j = len(height)-1
          while i<j:
              H = min(height[i],height[j])
              W = abs(i-j)
              maxarea = max(maxarea,H*W)
              if height[i]<height[j]:
                  i = i+1
              else:
                  j = j-1
          return maxarea
  ```
  ```