

250713盛最多水的容器（题解）

题目链接：

<https://leetcode.cn/problems/container-with-most-water/description/>

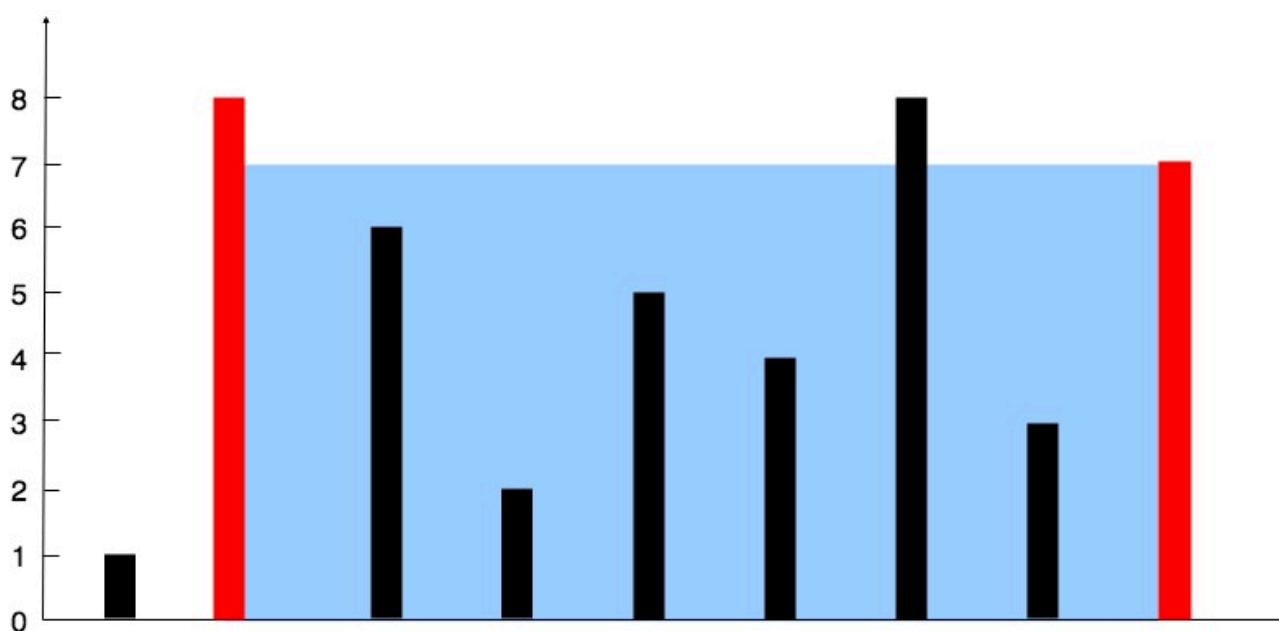
给定一个长度为 n 的整数数组 `height`。有 n 条垂线，第 i 条线的两个端点是 $(i, 0)$ 和 $(i, \text{height}[i])$ 。

找出其中的两条线，使得它们与 x 轴共同构成的容器可以容纳最多的水。

返回容器可以储存的最大水量。

说明：你不能倾斜容器。

示例 1：



输入：[1, 8, 6, 2, 5, 4, 8, 3, 7]

输出：49

解释：图中垂直线代表输入数组 [1, 8, 6, 2, 5, 4, 8, 3, 7]。在此情况下，容器能够容纳水（表示为蓝色部分）的最大值为 49。

解法一：双指针

1. `left` 指向最左边，`right` 指向最右边
2. 每次计算当前这两条线围成的面积：
3. 更新最大值 `ans`
4. 移动较短边：

- 如果 `height[left] < height[right]` , 说明左边短, 不可能再通过这个左边的线获得更大面积, 所以移动 `left++`
- 否则移动 `right--`

5. 重复直到 `left >= right`

为什么移动短的那一边?

- 高度是取决于 **较短的那一条线**
- 如果移动较高的那一边, **新的面积不会更大**, 因为高度不变或更低, 宽度还缩小了
- 所以只移动较短的那一边, 才有可能获得更大的面积

举个例子演示

以输入 `[1,8,6,2,5,4,8,3,7]` 为例:

- 初始: `left = 0` , `right = 8` , 高度 1 和 7 , 面积 `1*8=8`
- 移动左指针 (因为 `1 < 7`)
- 然后 `left = 1` , `right = 8` , 高度 8 和 7 , 面积 `7*7=49` (当前最大)
- 然后不断尝试更新, 最终最大面积为 **49**

```
class Solution {
public:
    int maxArea(vector<int>& height) {
        int ans = 0, left = 0, right = height.size() - 1;
        while (left < right) {
            int area = (right - left) * min(height[left], height[right]);
            ans = max(ans, area);
            height[left] < height[right] ? left++ : right--;
        }
        return ans;
    }
};
```