

11. 盛最多水的容器

在每个状态下，无论长板或短板向中间收窄一格，都会导致水槽 **底边宽度** -1 变短：

- 若向内 **移动短板**，水槽的短板 $\min(h[i], h[j])$ 可能变大，因此下个水槽的面积 **可能增大**。
- 若向内 **移动长板**，水槽的短板 $\min(h[i], h[j])$ 不变或变小，因此下个水槽的面积 **一定变小**。

因此，初始化双指针分列水槽左右两端，循环每轮将短板向内移动一格，并更新面积最大值，直到两指针相遇时跳出；即可获得最大面积。

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

$$S(i, j) = \min(h[i], h[j]) \times (j - i)$$

283. 移动零

```
class Solution {
public:
    void moveZeroes(vector<int> &nums) {
        int i = 0, j = 0;
        while (j < nums.size()) {
            if (nums[j]) {
                swap(nums[i++], nums[j]);
            }
            j++;
        }
    }
};
```

