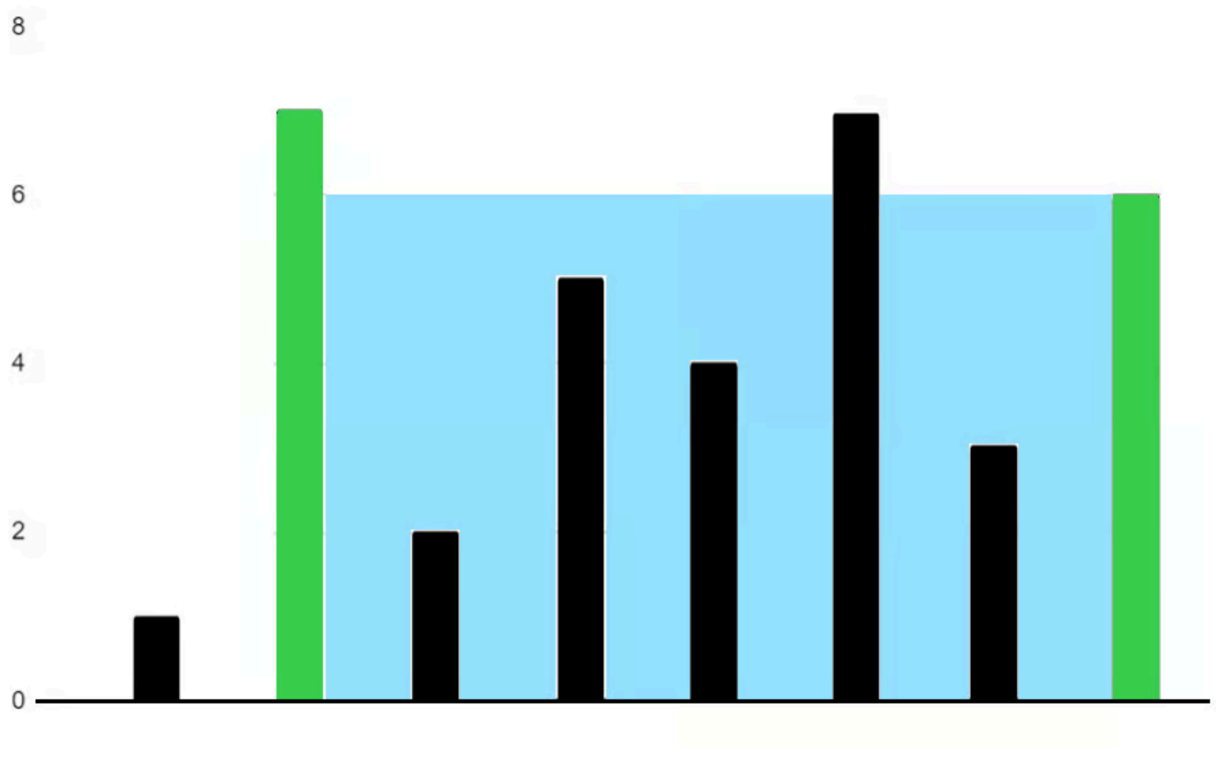
11. Container With Most Water (Medium)

You are given an integer array `heights` where `heights[i]` represents the height of the ith*ith* bar.

You may choose any two bars to form a container. Return the *maximum* amount of water a container can store.

Example 1:



```
Input: height = [1,7,2,5,4,7,3,6]

Output: 36
```

Example 2:

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

Output: 4
```

Constraints:

- `2 <= height.length <= 1000`
- `0 <= height[i] <= 1000`

思路: 大小取決於最小height，所以 two pointer只移動小的那個

做法

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