

# TWO POINTERS

# Problem – 11. Container With Most Water

Medium



LeetCode

[leetcode.com/problems/container-with-most-water](https://leetcode.com/problems/container-with-most-water)

## Problem Statement

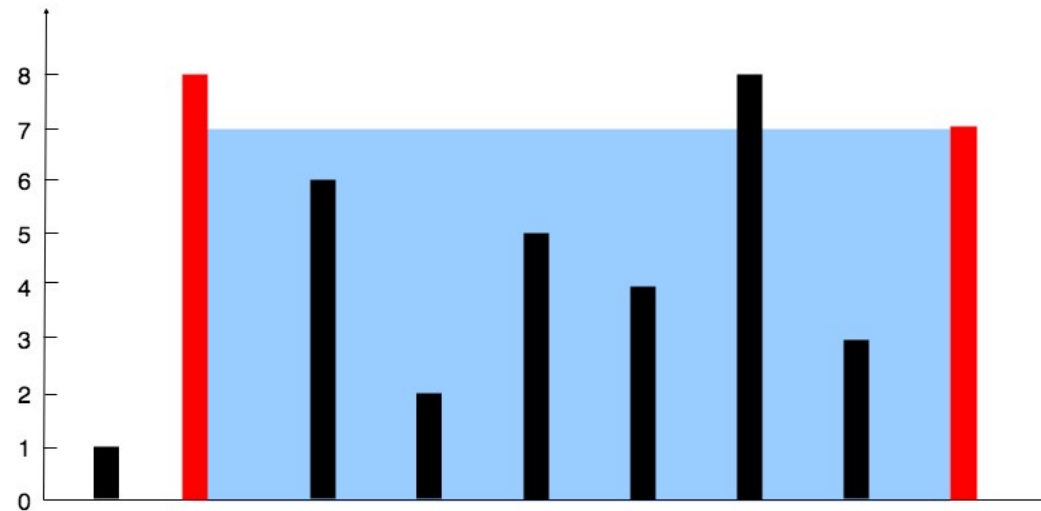
- You are given an integer array **height**
- Find two lines that together with x-axis form a container with most water
- Example:

### Input:

height = [1,8,6,2,5,4,8,3,7]

### Output:

49



# Solution – Container With Most Water

Medium



LeetCode

[leetcode.com/problems/container-with-most-water](https://leetcode.com/problems/container-with-most-water)

## Solution

- Initialize the maximum area **maxArea = 0**
- Initialize two pointers, **left = 0** and **right = height.size - 1**
- Loop while pointer **left < right**
- Calculate the area:  
**area = min(height[left], height[right]) \* (right - left)**
- Update the global maximum area:  
**maxArea = max(maxArea, area)**
- Move the smallest pointer (increment **left** or decrement **right**)
- Return **maxArea**

# Code – Container With Most Water

Medium



LeetCode

[leetcode.com/problems/container-with-most-water](https://leetcode.com/problems/container-with-most-water)

## Code

```
int maxArea(vector<int>& height) {  
    // initialize the two pointers (left and right)  
    int left = 0;  
    int right = height.size() - 1;  
    int maxArea = 0;  
    while (left < right) {  
        // calculate the area, think about the x-axis and y-axis  
        int area = min(height[left], height[right]) * (right - left);  
        // update maximum area  
        maxArea = max(area, maxArea);  
        // is the left pointer (y) smaller than right?  
        if (height[left] < height[right]) {  
            // move left pointer to right  
            left++;  
        } else {  
            // otherwise, move right pointer to left  
            right--;  
        }  
    }  
    return maxArea;  
}
```