# 11 盛最多水的容器

Label：双指针

给你 n 个非负整数 a1，a2，...，an，每个数代表坐标中的一个点 (i, ai) 。在坐标内画 n 条垂直线，垂直线 i 的两个端点分别为 (i, ai) 和 (i, 0) 。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。

说明：你不能倾斜容器。

- 双重循环（超出时间限制）

```
class Solution {
    public int maxArea(int[] height) {
        // 双重循环
        int[] max = new int[height.length];
        for (int i = 0; i < height.length; i++) {

            for (int j = i + 1; j < height.length; j++) {
                max[i] =Math.max(max[i], Math.min(height[i], height[j]) * (j -
i));
            }
        }

        return Arrays.stream(max).max().getAsInt();
    }
}
```

- 双指针

```
class Solution {
    public int maxArea(int[] height) {
        int max = 0;
        int end = height.length - 1;
        for(int start = 0; start < end ; ){
            int minHeight = height[start] < height[end] ? height[start++] :
height[end--];
            max = Math.max(max, (end - start + 1) * minHeight); // + 1是补上一行向
中间靠拢了1
        }

        return max;
    }
}
```

- 双指针（一开始两个指针一个指向开头一个指向结尾，此时容器的底是最大的，接下来随着指针向内移动，会造成容器的底变小，在这种情况下想要让容器盛水变多，就只有最大化容器的高）

```
class Solution {
    public int maxArea(int[] height) {
        int size = height.length;
        int left=0, right=size-1;
        int ans = 0;

        while(left < right){
            ans = Math.max(ans, (right - left)* Math.min(height[left],
height[right]));

            if(height[left] > height[right])
                right--;
            else
                left++;
        }
        return ans;
    }
}
```