Problem | Editorial | Submissions

Container With Most Water

Difficulty: **Medium** Accuracy: **53.84%** Submissions: **75K+** Points: **4**

Given an array **arr[]** of non-negative integers, where each element arr[i] represents the height of the **vertical lines,** find the **maximum amount of water** that can be contained between any two lines, together with the x-axis.

Note: In the case of a single vertical line it will not be able to hold water.

Examples:

Input: arr[] = [1, 5, 4, 3]
Output: 6
Explanation: 5 and 3 are 2 distance apart. So the size of the base is 2. Height of container = min(5, 3) = 3. So, total area to hold water = 3 * 2 = 6.

Input: arr[] = [3, 1, 2, 4, 5]
Output: 12
Explanation: 5 and 3 are 4 distance apart. So the size of the base is 4. Height of container = min(5, 3) = 3. So, total area to hold water = 4 * 3 = 12.

Java (1.8) | Average Time: 30m | Start Timer

```
1  // } Driver Code Ends
19 class Solution {
20     public int maxWater(int arr[]) {
21         int l = 0, r = arr.length - 1, maz = 0;
22         while (l < r) {
23             int h = Math.min(arr[l], arr[r]);
24             int w = r - l;
25             int ca = h * w;
26             maz = Math.max(maz, ca);
27             if (arr[l] <= arr[r]) {
28                 l++;
29             } else {
30                 r--;
31             }
32         }
33         return maz;
34     }
35 }
```

Custom Input | Compile & Run | Submit