

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(n) time | O(n) space
6 func WaterArea(heights []int) int {
7     maxes := make([]int, len(heights))
8     leftmax := 0
9     for i, height := range heights {
10         maxes[i] = leftmax
11         leftmax = max(leftmax, height)
12     }
13     rightmax := 0
14     for i := range heights {
15         j := len(heights) - i - 1
16         height := heights[j]
17         minheight := min(rightmax, maxes[j])
18         if height < minheight {
19             maxes[j] = minheight - height
20         } else {
21             maxes[j] = 0
22         }
23         rightmax = max(rightmax, height)
24     }
25     return sum(maxes)
26 }
27
28 func min(arg1 int, rest ...int) int {
29     curr := arg1
30     for _, num := range rest {
31         if num < curr {
32             curr = num
33         }
34     }
35     return curr
36 }
37
38 func max(arg1 int, rest ...int) int {
39     curr := arg1
40     for _, num := range rest {
41         if num > curr {
42             curr = num
43         }
44     }
45     return curr
46 }
47
48 func sum(arr []int) int {
49     acc := 0
50     for _, num := range arr {
51         acc += num
52     }
53     return acc
54 }
55
```