

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n) time | O(n) space
5     func waterArea(heights: [Int]) -> Int {
6         var maxes = Array(repeating: 0, count: heights.count)
7         var leftMax = 0
8
9         for i in 0 ..< heights.count {
10             let currentHeight = heights[i]
11
12             maxes[i] = leftMax
13             leftMax = max(leftMax, currentHeight)
14         }
15
16         var rightMax = 0
17
18         for i in (0 ..< heights.count).reversed() {
19             let currentHeight = heights[i]
20
21             let minMax = min(rightMax, maxes[i])
22
23             if currentHeight < minMax {
24                 maxes[i] = minMax - currentHeight
25             } else {
26                 maxes[i] = 0
27             }
28
29             rightMax = max(rightMax, currentHeight)
30         }
31
32         return maxes.reduce(0) {
33             x, y in
```

Solution 1   Solution 2   Solution 3

```
1 class Program {
2     func waterArea(heights: [Int]) -> Int {
3         // Write your code here.
4         return -1
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func waterArea(heights: [Int]) -> Int {
3         // Write your code here.
4         return -1
5     }
6 }
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n) time | O(n) space
5     func waterArea(heights: [Int]) -> Int {
6         var maxes = Array(repeating: 0, count: heights.count)
7         var leftMax = 0
8
9         for i in 0 ..< heights.count {
10             let currentHeight = heights[i]
11
12             maxes[i] = leftMax
13             leftMax = max(leftMax, currentHeight)
14         }
15
16         var rightMax = 0
17
18         for i in (0 ..< heights.count).reversed() {
19             let currentHeight = heights[i]
20
21             let minMax = min(rightMax, maxes[i])
22
23             if currentHeight < minMax {
24                 maxes[i] = minMax - currentHeight
25             } else {
26                 maxes[i] = 0
27             }
28
29             rightMax = max(rightMax, currentHeight)
30         }
31
32         return maxes.reduce(0) {
33             x, y in
```

```

18         self.eventQueue.put(self.program.adjustWindowSize( 20, 10, 10, 10 ))
19     }
20     self.start("Test Case 4") <= 10 seconds <= self.end()
21     self.eventQueue.put(self.program.adjustWindowSize( 20, 10, 10, 10 ))
22 }
23 self.start("Test Case 4") <= 10 seconds <= self.end()
24 self.eventQueue.put(self.program.adjustWindowSize( 20, 10, 10, 10 ))
25 }
26 self.start("Test Case 4") <= 10 seconds <= self.end()
27 self.eventQueue.put(self.program.adjustWindowSize( 20, 10, 10, 10 ))
28 }
29 self.start("Test Case 4") <= 10 seconds <= self.end()
30 self.eventQueue.put(self.program.adjustWindowSize( 20, 10, 10, 10 ))
31 }
32 self.start("Test Case 4") <= 10 seconds <= self.end()

```

Run or submit code when you're ready.