Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(n) time | O(n) space
  public static int waterArea(int[] heights) {
    int[] maxes = new int[heights.length];
    int leftMax = 0;
    for (int i = 0; i < heights.length; i++) {
      int height = heights[i];
      maxes[i] = leftMax;
      leftMax = Math.max(leftMax, height);
    }
    int rightMax = 0;
    for (int i = heights.length - 1; i >= 0; i--) {
      int height = heights[i];
      int minHeight = Math.min(rightMax, maxes[i]);
      if (height < minHeight) {
        maxes[i] = minHeight - height;
      } else {
        maxes[i] = 0;
      }
      rightMax = Math.max(rightMax, height);
    }
    int total = 0;
    for (int i = 0; i < heights.length; i++) {
      total += maxes[i];
    }
    return total;
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static int waterArea(int[] heights) {
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

```
@Test
public void Test3Level() {
    int[] input = {0, 0, 0, 0, 0};
    Utils.assertTrue(Program.sabotage(input) == 0);
}

@Test
public void Test4Level() {
    int[] input = {0, 1, 0, 0, 0};
    Utils.assertTrue(Program.sabotage(input) == 0);
}

@Test
public void Test5Level() {
    int[] input = {0, 1, 1, 0, 0};
```

Run or submit code when you're ready.