Our Solution(s) Run Code

Solution 1 Solution 2

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

class Program {
  // O(n) time | O(1) space - where n is the length of the array
  public static boolean isMonotonic(int[] array) {
    if (array.length <= 2) return true;

    var direction = array[1] - array[0];
    for (int i = 2; i < array.length; i++) {
      if (direction == 0) {
        direction = array[i] - array[i - 1];
        continue;
      }

      if (breaksDirection(direction, array[i - 1], array[i])) {
        return false;
      }
    }
    return true;
  }

  public static boolean breaksDirection(int direction, int previ
    var difference = current - previous;
    if (direction > 0) return difference < 0;
    return difference > 0;
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static boolean isMonotonic(int[] array) {
    // Write your code here.
    return false;
  }
}

```

Our Tests

Custom Output Submit Code

```
@Test
public void Test1(){
  var input = new int[] {},
  var expected = true,
  var actual = Program.IsMonotonic(input),
  Utils.assertEquals(expected, actual),
}

@Test
public void Test2(){
  var input = new int[] {6},
  var expected = true,
  var actual = Program.IsMonotonic(input),
  Utils.assertEquals(expected, actual),
}

@Test
public void Test3(){
  var input = new int[] {1, 2},
  var expected = true,
  var actual = Program.IsMonotonic(input),
```

Run or submit code when you're ready.