Three Number Sum

Input: array = [-1, -5, -10, -1100, -1100, -1101, -1102, -9001]

Output: true

```
// O(n) time | O(1) space
function isMonotonic(array) {
  let firstPointer = 0;
  let secondPointer = 1;
  let initalSign = Math.sign(array[secondPointer] - array[firstPointer]);
  while (initalSign === 0) {
    firstPointer++;
    secondPointer++;
    initalSign = Math.sign(array[secondPointer] - array[firstPointer]);
  }
  let newSign;

  for (let i = firstPointer; i < array.length - 1; i++) {
    firstPointer = i;
    secondPointer = i + 1;
    newSign = Math.sign(array[secondPointer] - array[firstPointer]);
    if (newSign !== initalSign && newSign !== 0) {
      return false;
    }
  }
  return true;
}
```

Input: An array of integers

Output: A boolean representing whether the array is monotonic

// monotonic: if its elements, from L to R, are entirely non-increasing or entirely non-decreasing

Time: O(n) (where n is the length of the input) since we will be looping through every element of the array

Space: O(1) since we are not using any more space as the input size grows

Idea: Use two pointers that get compared at each iteration

Mistake #1: Initial while loop. We could have the first two elements be the same therefore, we must increment our pointers until they point at different elements

Mistake #2: AND not OR. If our new sign is not equal to the initial sign AND it isn't equal to zero that means it is not monotonic. OR doesn't make sense