

Three Number Sum

Input: arrayOne = [-1, 5, 10, 20, 28, 3]
arrayTwo = [26, 134, 135, 15, 17]

Output: [28, 26]

Input: Two non-empty arrays of integers

Output: An array containing the pair of numbers (one from each array) whose absolute difference closest to zero
Number from the first array should be in the first position

```
// O(nlogn + mlogm) time | O(1) space
function smallestDifference(arrayOne, arrayTwo) {
  arrayOne.sort((a, b) => a - b);
  arrayTwo.sort((a, b) => a - b);
  let arrayOnePointer = 0;
  let arrayTwoPointer = 0;
  let absDifference = Infinity;
  let arrayOneWinner = null;
  let arrayTwoWinner = null;
  let currAbsDifference = Infinity;

  while (
    arrayOnePointer < arrayOne.length &&
    arrayTwoPointer < arrayTwo.length
  ) {
    let firstNum = arrayOne[arrayOnePointer];
    let secondNum = arrayTwo[arrayTwoPointer];

    if (firstNum === secondNum) {
      return [firstNum, secondNum];
    } else if (firstNum < secondNum) {
      currAbsDifference = secondNum - firstNum;
      arrayOnePointer++;
    } else if (firstNum > secondNum) {
      currAbsDifference = firstNum - secondNum;
      arrayTwoPointer++;
    }

    if (currAbsDifference < absDifference) {
      absDifference = currAbsDifference;
      arrayOneWinner = firstNum;
      arrayTwoWinner = secondNum;
    }
  }

  return [arrayOneWinner, arrayTwoWinner];
}
```

Time: $O(n \log n + m \log m)$ where n and m are the lengths of arrayOne and arrayTwo respectively

Space: $O(1)$ since we are using a fixed with array to return our winners and there are no space implications as input size grows

I initially had OR. This is wrong because OR is still true if one value is true and one is false. We need AND since we want to end the loop when we surpass the length of even one array

We always increment the smallest value. This is because the array is sorted and incrementing the smallest # of the two will move us to a smaller difference