Three Number Sum

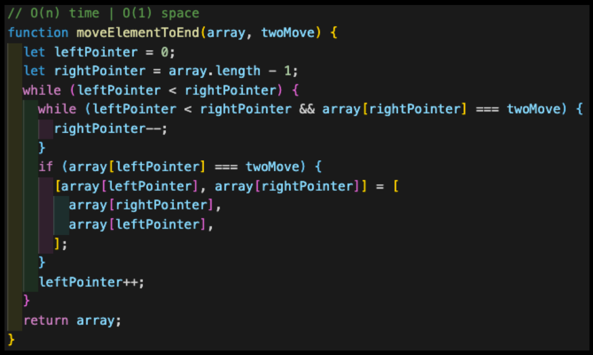
Input: array = [2, 1, 2, 2, 2, 3, 4, 2]
toMove = 2

Output: [1, 3, 4, 2, 2, 2, 2, 2]

// 1, 3, 4 could be ordered differently

Input: An array of integers and
An integer

Output: An array where all instances of the given input integer in the array is moved to the end of the array



```
// O(n) time | O(1) space
function moveElementToEnd(array, twoMove) {
  let leftPointer = 0;
  let rightPointer = array.length - 1;
  while (leftPointer < rightPointer) {
    while (leftPointer < rightPointer && array[rightPointer] === twoMove) {
      rightPointer--;
    }
    if (array[leftPointer] === twoMove) {
      [array[leftPointer], array[rightPointer]] = [
        array[rightPointer],
        array[leftPointer],
      ];
    }
    leftPointer++;
  }
  return array;
}
```

Time: O(n) (where n is the length of the input array) since we loop over every element of the input array

Space: O(1) since we do not store any values, we just swap them

Idea: Have two pointers at either end of the array. We swap the values at these pointers with each other based on certain conditions

Mistake #1: Second while loop. We need a while loop here so that we can keep decrementing the rightPointer until !== toMov. We must check if leftPointer < rightPointer at each iteration since we are decrementing the rightPointer

Mistake #2 leftPointer++. We should move it to the end of our logic bc in any case, we always want to increment the leftPointer after an iteration