Beaumont Yin
CS 146

Task: Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Approach:

1. Sort the given array to preform two pointer search
2. Skip the duplicates when nums[i]==nums[i-1]
3. For each value of i make two pointers called left and right
4. Take the sum of these elements
5. If the sum is zero then add the triplets to the result
6. Update pointers left and right to make sure they are in bounds then repeat steps 2-6 until you have found every correct combination

Test cases:

```
Solution solution = new Solution();
        int[] nums1 = {-1, 0, 1, 2, -1, -4};
        System.out.println(solution.threeSum(nums1));

        int[] nums2 = {0, 1, 1};
        System.out.println(solution.threeSum(nums2));

        int[] nums3 = {0, 0, 0};
        System.out.println(solution.threeSum(nums3));
```