LeetCode 15

Description

3 Sum
Given an array nums of n integers, are there elements a, b, c in nums such that a + b + c = 0? Find all unique triplets in the array which gives the sum of zero.

Note:

The solution set must not contain duplicate triplets.

Example:

Given array nums = [-1, 0, 1, 2, -1, -4],

A solution set is:
[
[-1, 0, 1],
[-1, -1, 2]
]

Thought

First we can think of the way we solve 2sum and 3sum cloest, we fix one element, and create two pointers, one in the front and one in the end of the rest of elements. Then we check its sum's difference with the target.

If sum > tgt, we move the end pointer left
If sum < tgt, we move the start pointer right
If sum == tgt, we add it to the result

We apply the same process to each fixed element in the loop.

Solution

```
public List<List<Integer>> threeSum(int[] num) {
    Arrays.sort(num);
    List<List<Integer>> res = new LinkedList<>();
    for (int i = 0; i < num.length-2; i++) {
        if (i == 0 || (i > 0 && num[i] != num[i-1])) {
            int lo = i+1, hi = num.length-1, sum = 0 - num[i];
            while (lo < hi) {
                if (num[lo] + num[hi] == sum) {
                    res.add(Arrays.asList(num[i], num[lo], num[hi]));
                    while (lo < hi && num[lo] == num[lo+1]) lo++;
                    while (lo < hi && num[hi] == num[hi-1]) hi--;
                    lo++; hi--;
                } else if (num[lo] + num[hi] < sum) lo++;
                else hi--;
           }
        }
    }
    return res;
}
```

Takeaways

- 3sum break into 2sum
- mind the edge cases