Problem no:15 3 sum

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.

Notice that the solution set must not contain duplicate triplets.

**Example 1:**

**Input:** nums = [-1,0,1,2,-1,-4]

**Output:** [[-1,-1,2],[-1,0,1]]

**Explanation:**

nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.

nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.

nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.

The distinct triplets are [-1,0,1] and [-1,-1,2].

Notice that the order of the output and the order of the triplets does not matter.

**Example 2:**

**Input:** nums = [0,1,1]

**Output:** []

**Explanation:** The only possible triplet does not sum up to 0.

**Example 3:**

**Input:** nums = [0,0,0]

**Output:** [[0,0,0]]

**Explanation:** The only possible triplet sums up to 0.

**Constraints:**

* 3 <= nums.length <= 3000
* -105 <= nums[i] <= 105

JAVA PROGRAM

class Solution {

    public List<List<Integer>> threeSum(int[] nums) {

         int n = nums.length;

        List<List<Integer>> res = new ArrayList<>();

        Arrays.sort(nums);

        for(int i=0;i<n;i++){

            if(i>0 && nums[i]==nums[i-1]) continue;

            int j = i+1;

            int k = n-1;

            while(j<k){

                int sum = nums[i] + nums[j] + nums[k];

                if(sum<0) j++;

                else if(sum>0) k--;

                else{

                      ArrayList<Integer> temp = new ArrayList<>();

                        temp.add(nums[i]);

                        temp.add(nums[j]);

                        temp.add(nums[k]);

                        res.add(temp);

                        j++;

                        k--;

                        while(j<k && nums[j]==nums[j-1]) j++;

                        while(j<k && nums[k]==nums[k+1]) k--;

                }

            }

        }

        return res;

    }

}

OUTPUT

Input

nums =

[-1,0,1,2,-1,-4]

Output

[[-1,-1,2],[-1,0,1]]

Expected

[[-1,-1,2],[-1,0,1]]