JS

function search\_triplets(arr) {

arr.sort((a, b) => a - b);

const triplets = [];

for (i = 0; i < arr.length; i++) {

if (i > 0 && arr[i] === arr[i - 1]) { // skip same element to avoid duplicate triplets

continue;

}

search\_pair(arr, -arr[i], i + 1, triplets);

}

return triplets;

}

function search\_pair(arr, target\_sum, left, triplets) {

let right = arr.length - 1;

while (left < right) {

const current\_sum = arr[left] + arr[right];

if (current\_sum === target\_sum) { // found the triplet

triplets.push([-target\_sum, arr[left], arr[right]]);

left += 1;

right -= 1;

while (left < right && arr[left] === arr[left - 1]) {

left += 1; // skip same element to avoid duplicate triplets

}

while (left < right && arr[right] === arr[right + 1]) {

right -= 1; // skip same element to avoid duplicate triplets

}

} else if (target\_sum > current\_sum) {

left += 1; // we need a pair with a bigger sum

} else {

right -= 1; // we need a pair with a smaller sum

}

}

}

console.log(search\_triplets([-3, 0, 1, 2, -1, 1, -2]));

console.log(search\_triplets([-5, 2, -1, -2, 3]));

Java

import java.util.\*;

class TripletSumToZero {

public static List<List<Integer>> searchTriplets(int[] arr) {

Arrays.sort(arr);

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

for (int i = 0; i < arr.length - 2; i++) {

if (i > 0 && arr[i] == arr[i - 1]) // skip same element to avoid duplicate triplets

continue;

searchPair(arr, -arr[i], i + 1, triplets);

}

return triplets;

}

private static void searchPair(int[] arr, int targetSum, int left, List<List<Integer>> triplets) {

int right = arr.length - 1;

while (left < right) {

int currentSum = arr[left] + arr[right];

if (currentSum == targetSum) { // found the triplet

triplets.add(Arrays.asList(-targetSum, arr[left], arr[right]));

left++;

right--;

while (left < right && arr[left] == arr[left - 1])

left++; // skip same element to avoid duplicate triplets

while (left < right && arr[right] == arr[right + 1])

right--; // skip same element to avoid duplicate triplets

} else if (targetSum > currentSum)

left++; // we need a pair with a bigger sum

else

right--; // we need a pair with a smaller sum

}

}

public static void main(String[] args) {

System.out.println(TripletSumToZero.searchTriplets(new int[] { -3, 0, 1, 2, -1, 1, -2 }));

System.out.println(TripletSumToZero.searchTriplets(new int[] { -5, 2, -1, -2, 3 }));

}

}