JS

function triplet\_sum\_close\_to\_target(arr, targetSum) {

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

let smallest\_difference = Infinity;

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

let left = i + 1,

right = arr.length - 1;

while (left < right) {

const target\_diff = targetSum - arr[i] - arr[left] - arr[right];

if (target\_diff === 0) { // we've found a triplet with an exact sum

return targetSum - target\_diff; // return sum of all the numbers

}

if (Math.abs(target\_diff) < Math.abs(smallest\_difference)) {

smallest\_difference = target\_diff; // save the closest difference

}

// the second part of the following 'if' is to handle the smallest sum when we have more than one solution

if (Math.abs(target\_diff) < Math.abs(smallest\_difference) ||

(Math.abs(target\_diff) === Math.abs(smallest\_difference) && target\_diff > smallest\_difference)) {

smallest\_difference = target\_diff; // save the closest and the biggest difference

}

if (target\_diff > 0) {

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

} else {

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

}

}

}

return targetSum - smallest\_difference;

}

console.log(triplet\_sum\_close\_to\_target([-2, 0, 1, 2], 2));

console.log(triplet\_sum\_close\_to\_target([-3, -1, 1, 2], 1));

console.log(triplet\_sum\_close\_to\_target([1, 0, 1, 1], 100));

Java

import java.util.\*;

class TripletSumCloseToTarget {

public static int searchTriplet(int[] arr, int targetSum) {

if (arr == null || arr.length < 3)

throw new IllegalArgumentException();

Arrays.sort(arr);

int smallestDifference = Integer.MAX\_VALUE;

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

int left = i + 1, right = arr.length - 1;

while (left < right) {

// comparing the sum of three numbers to the 'targetSum' can cause overflow

// so, we will try to find a target difference

int targetDiff = targetSum - arr[i] - arr[left] - arr[right];

if (targetDiff == 0) // we've found a triplet with an exact sum

return targetSum - targetDiff; // return sum of all the numbers

// the second part of the above 'if' is to handle the smallest sum when we have more than one solution

if (Math.abs(targetDiff) < Math.abs(smallestDifference)

|| (Math.abs(targetDiff) == Math.abs(smallestDifference) && targetDiff > smallestDifference))

smallestDifference = targetDiff; // save the closest and the biggest difference

if (targetDiff > 0)

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

else

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

}

}

return targetSum - smallestDifference;

}

public static void main(String[] args) {

System.out.println(TripletSumCloseToTarget.searchTriplet(new int[] { -2, 0, 1, 2 }, 2));

System.out.println(TripletSumCloseToTarget.searchTriplet(new int[] { -3, -1, 1, 2 }, 1));

System.out.println(TripletSumCloseToTarget.searchTriplet(new int[] { 1, 0, 1, 1 }, 100));

}

}