Java-排序数组-给定一个随机的整数

题目描述：

给定一个随机的整数（可能存在正整数和负整数）数组 nums ，请你在该数组中找出两个数，其和的绝对值(|nums[x]+nums[y]|)为最小值，并返回这个两个数（按从小到大返回）以及绝对值。

每种输入只会对应一个答案。但是，数组中同一个元素不能使用两遍。

输入描述：

一个通过空格分割的有序整数序列字符串，最多 1000 个整数，且整数数值范围是 [-65535, 65535]。

输出描述：

两数之和绝对值最小值

补充说明

示例 1

输入：

-1 -3 7 5 11 15

输出：

-3 5 2

说明：

因为 |nums[0] + nums[2]| = |-3 + 5| = 2 最小，所以返回 -3 5 2

```
import java.util.Scanner;
import java.util.Arrays;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s = in.nextLine();
        String[] split = s.split(" ");
        int[] arr = new int[split.length];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = Integer.parseInt(split[i]);
        }
//          int[] arr = {-1, -3, 7, 5, 11, 15, 4, 6, 4, 87, 6, 3, 1, 2, 3};
        getMinAbs(arr);
    }

    private static void getMinAbs(int[] arr) {
        if (arr == null || arr.length < 2) {
            return;
        }

        Arrays.sort(arr);
        // 全大于 0
        if (arr[0] >= 0) {
```

```
            System.out.println(arr[0] + " " + arr[1] + " " + (arr[0] + arr[1]));
            return;
        }
        // 全小于 0
        if (arr[arr.length - 1] <= 0) {
            System.out.println(arr[arr.length - 2] + " " + arr[arr.length - 1] + " " +
Math.abs(arr[arr.length - 2] + arr[arr.length - 1]));
            return;
        }
        int index = -1;
        // 大于 0，小于 0 都有
        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] < 0 && arr[i + 1] >= 0) {
                index = i;
                break;
            }
        }

        int resLeft = 0;
        int resRight = 0;
        int resSum = Integer.MAX_VALUE;
        for (int i = index + 1; i < arr.length; i++) {
            int minIndex = findMinIndex(i, arr, index);
            if (Math.abs(arr[minIndex] + arr[i]) < resSum) {
                resSum = Math.abs(arr[minIndex] + arr[i]);
                resLeft = minIndex;
                resRight = i;
            }
        }
        System.out.println(arr[resLeft] + " " + arr[resRight] + " " + resSum);
    }

    private static int findMinIndex(int curIndex, int[] arr, int rightSide) {
        int left = 0;
        int right = rightSide;
        while (left < right) {
            int mid = (right - left) / 2;
            int t = arr[mid] + arr[curIndex];
            if (t > 0) {
                right = mid - 1;
            } else if (t < 0) {
                left = mid + 1;
            } else {
                return mid;
```

```
                }
            }
            if (left < 0 || left > rightSide) {
                return right;
            }
            if (right < 0 || right > rightSide) {
                return left;
            }
            return Math.abs(arr[left] + arr[curIndex]) >= Math.abs(arr[right] +
                    arr[curIndex]) ? right : left;
        }
}
```