

乱序整数序列两数之和绝对值最小

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( $|\text{nums}[x] + \text{nums}[y]|$ )为最小值，并返回这两个数（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：

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( $|\text{nums}[x] + \text{nums}[y]|$ )为最小值，并返回这两个数（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：

```
import java.util.*;
```

```
// 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);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```

// while (in.hasNextInt()) { // 注意 while 处理多个 case
//     int a = in.nextInt();
//     int b = in.nextInt();
//     System.out.println(a + b);
// }
String[] strs = in.nextLine().split(" ");
int len = strs.length;
int[] nums = new int[len];
for (int i = 0; i < len; i++) {
    nums[i] = Integer.parseInt(strs[i]);
}
String result = findMinAbsSum(nums);
System.out.println(result);
}

public static String findMinAbsSum(int[] nums) {
    Arrays.sort(nums);
    int minAbsSum = Integer.MAX_VALUE;
    int[] result = new int[2];
    int left = 0, right = nums.length - 1;
    while (left < right) {
        int curSum = nums[left] + nums[right];
        int currentAbsSum = Math.abs(curSum);
        if (currentAbsSum < minAbsSum) {
            minAbsSum = currentAbsSum;
            result[0] = nums[left];
            result[1] = nums[right];
        }
        if (curSum < 0) {
            left++;
        } else if (curSum > 0) {
            right--;
        } else {
            return result[0] + " " + result[1] + " " + minAbsSum;
        }
    }
    return result[0] + " " + result[1] + " " + minAbsSum;
}
}

```