**问题描述**

|  |  |
| --- | --- |
| 试题编号： | 201612-1 |
| 试题名称： | 中间数 |
| 时间限制： | 1.0s |
| 内存限制： | 256.0MB |
| 问题描述： | **问题描述**  　　在一个整数序列*a*1, *a*2, …, *an*中，如果存在某个数，大于它的整数数量等于小于它的整数数量，则称其为中间数。在一个序列中，可能存在多个下标不相同的中间数，这些中间数的值是相同的。 　　给定一个整数序列，请找出这个整数序列的中间数的值。  **输入格式**  　　输入的第一行包含了一个整数*n*，表示整数序列中数的个数。 　　第二行包含*n*个正整数，依次表示*a*1, *a*2, …, *an*。  **输出格式**  　　如果约定序列的中间数存在，则输出中间数的值，否则输出-1表示不存在中间数。  **样例输入**  6 2 6 5 6 3 5  **样例输出**  5  **样例说明**  　　比5小的数有2个，比5大的数也有2个。  **样例输入**  4 3 4 6 7  **样例输出**  -1  **样例说明**  　　在序列中的4个数都不满足中间数的定义。  **样例输入**  5 3 4 6 6 7  **样例输出**  -1  **样例说明**  　　在序列中的5个数都不满足中间数的定义。  **评测用例规模与约定**  　　对于所有评测用例，1 ≤ *n* ≤ 1000，1 ≤ *ai* ≤ 1000。 |

思路：这道题看起来是要排序后直接输出，但好像还要加一些对左右两边的判断。

代码：

package cxz;

import java.text.DecimalFormat;

import java.io.\*;

import java.util.\*;

import java.math.\*;

public class cxz{

static Scanner in=new Scanner(System.in);

static int n=0;

static int[] num=new int[10000];

static int[] cobe=new int[10000];

public static void main(String[] args) {

n=in.nextInt();

for(int i=1;i<=n;++i) {

num[i]=in.nextInt();

}

Arrays.sort(num,1,n+1);

int count=0;

for(int i=1;i<=n;++i) {

if(num[i]==num[i-1]) {

cobe[count]++;

}else {

count++;

cobe[count]++;

}

}

int sum=0;

boolean cxz=false;

for(int i=1;i<=count;++i) {

if(2\*sum==n-cobe[i]) {

System.out.println(num[sum+1]);

cxz=true;

break;

}else {

sum+=cobe[i];

}

}

if(!cxz) {

System.out.println(-1);

}

}

}

