图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main {

public static void main(String[] args){

Scanner input = new Scanner(System.in);

int N = input.nextInt();

int[] a = new int[N];

for (int i = 0; i < N; ++i)

a[i] = input.nextInt();

Arrays.sort(a);

for (int i = 0; i < N; ++i)

System.out.print(a[i] + " ");

}

}

图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int []A=new int[3];

for(int i=0;i<3;++i)

A[i]=input.nextInt();

Arrays.sort(A);

for(int i=2;i>=0;--i)

System.out.printf("%d ",A[i]);

}

}

图片包含 屏幕截图

描述已自动生成

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner input=new Scanner(System.in);

ArrayList<Integer>list=new ArrayList<>();

for(int a=input.nextInt();a!=0;a=input.nextInt())//读取数据

list.add(a);

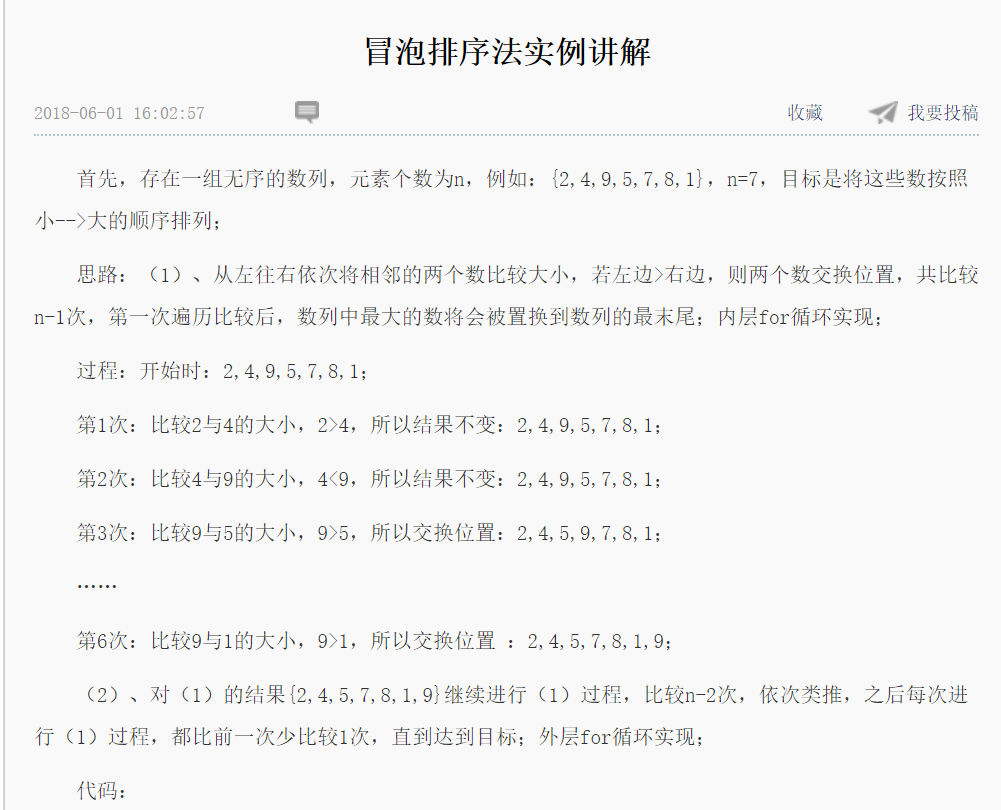
Collections.reverse(list);//将序列翻转

for(int i:list)

System.out.print(i+" ");

}

}



public class Main

{

    public static void main(String[] args)

    {

        int[] arr ={2,4,9,5,7,8,1};

        //排序

        for(int i =1; iarr[j+1])

                {

                    flag =false;

                    int temp =arr[j];

                    arr[j] =arr[j+1];

                    arr[j+1] =temp;

                }

           }

            //如果没有发生交换，说明已经排好序了，直接跳出循环

            if(flag)

            {

                break;

            }

        }

        System.out.println(Arrays.toString(arr));

    }

}

**实现二分查找算法：有序的数组**

输入你想安排的元素个数：

5

输入有序的元素+空格不断输入：

1 2 3 4 5

输入你想查询的数字key：

2

第1次二分查找范围0---------------4

第2次二分查找范围0---------------1

第3次二分查找范围1---------------1

与key相等的元素下表为：1

import java.util.Scanner;

public class BinarySearch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("输入你想安排的元素个数：");

int n = scanner.nextInt();//确定数组里放置几个元素

int[] arr = new int[n] //放置数组里的元素

System.out.println("输入有序的元素+空格不断输入：");

for (int i = 0; i < n; i++) {

arr[i] = scanner.nextInt();

}

System.out.println("输入你想查询的数字key：");

int key = scanner.nextInt();//输入要查询的数

int indexNum = binarySearch(arr, key);

System.out.println("与key相等的元素下表为：" + indexNum);

}

static int binarySearch(int[] arr, int a) {

int low = 0；//代表索引

int high = arr.length - 1;

int count = 1;//计数

while (low <= high) {

System.out.println("第" + count + "次二分查找范围" + low + "---------------" + high);

int mid = (low + high) / 2;

if (arr[mid] == a) {

return mid;

} else if (arr[mid] > a) {

high = mid - 1 ;//如果单纯为 high = mid，low = mid，查找最后一个数会陷入无限循环

} else {

low = mid + 1;

}

count++;

}

return -1;

}

}