

# Java方向编程题答案

day41

[编程题]: 58392 连续最大和

<https://www.nowcoder.com/questionTerminal/5a304c109a544aef9b583dce23f5f5db>

【题目解析】：题目意思是求解连续子数组的最大和比如：{1,4,-10,9,2,-1}。连续子数组最大和为：11。

【解题思路】：

思路一：暴力求解。当遍历到某个元素的时候，直接从当前数字开始进行累加后续的元素，每次把最大值保存下来。求得最后的解。但是，**时间复杂度高。达到了 $O(n^2)$ 。运行会超时。**

思路二：直接进行累加，与最大值比较，并不停地更替最大值。详情见【示例代码二】。

【示例代码一】：运行会超时

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = 0;
        while(sc.hasNext()){
            n = sc.nextInt();
            int[] num = new int[n];
            for(int i=0;i<n;i++){
                num[i] = sc.nextInt();
            }
            int result = maxArray(num,n);
            System.out.println(result);
        }
    }

    public static int maxArray(int[] arr,int len) {
        if(arr == null || len <= 0){
            return -1;
        }
        int maxnum = 0x80000000;
        int sum;
        for(int i = 0;i < len;i++)
        {
            //每次从当前位置i开始进行累加，求最大值。
            sum = 0;
            for(int j = i;j < len;j++)
            {
                sum = sum+arr[j];
                if(sum > maxnum)
                {
                    maxnum = sum;
                }
            }
        }
    }
}
```

```

    }
    return maxnum;
}
}

```

【示例代码二】：

```

import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = 0;
        while(sc.hasNext()){
            n = sc.nextInt();
            int[] num = new int[n];
            for(int i=0;i<n;i++){
                num[i] = sc.nextInt();
            }
            int result = maxArray(num,n);
            System.out.println(result);
        }
    }

    public static int maxArray(int[] array,int len) {
        if(array == null || len <= 0) {
            return -1;
        }
        //假设最大值为这个数字，这里有一个几千，这个数字正好是整形的最小值。
        int maxnum = 0x80000000;
        int sum = 0;
        for(int i = 0;i < len;i++) {
            if(sum <= 0) {
                sum = array[i];
            }else{
                sum += array[i];
            }
            if(sum > maxnum) {
                maxnum = sum;
            }
        }
        return maxnum;
    }
}

```

[编程题]36841-坐标移动

<https://www.nowcoder.com/questionTerminal/119bcc3befb405f58abe9c532eb29>

**【题目解析】**：这个题的意思其实就是:合法坐标为A(或者D或者W或者S) + 数字（两位以内）。坐标之间以;分隔。非法坐标点需要进行丢弃。如AA10; A1A; ; YAD; 等。然后给定了一个字符串：A10;S20;W10;D30;X;A1A;B10A11;;A10;。对这个字符串进行操作。

**【解题思路】：** 先将整个字符串用分号";"进行分割，这样就把每一个坐标给取出来了。接下来对取出的每个坐标进行合法性判断和计算，这里注意，我使用了正则表达式。"[0-9]{1,2}"表示1~2个数字.如果是：[0-9]+ 表示：表示至少1个、最多不限制的数字串。

**【示例代码】：**

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        //判断是否有下一个录入的元素
        while(sc.hasNext()){
            String str=sc.nextLine();
            //使用;切割录入的字符串数据,返回字符串数组
            String[] A=str.split(";");
            int x=0,y=0;
            for(String string:A){
                //遍历字符串数组获取每一个部分内容,分别判断0索引对应的元素的内容
                //判断从1号位置开始是否有连续的1到2位的数字。(数字是0-9)
                if(string.charAt(0)=='D' && string.substring(1).matches("[0-9]{1,2}"))
                    //将字符串的字符转换为整形数字
                    x+=Integer.parseInt(string.substring(1));

                if(string.charAt(0)=='W' && string.substring(1).matches("[0-9]{1,2}"))
                    y+=Integer.parseInt(string.substring(1));

                if(string.charAt(0)=='S' && string.substring(1).matches("[0-9]{1,2}"))
                    y-=Integer.parseInt(string.substring(1));

                if(string.charAt(0)=='A' && string.substring(1).matches("[0-9]{1,2}"))
                    x-=Integer.parseInt(string.substring(1));
            }
            System.out.println(x+","+y);
        }
        sc.close();
    }
}
```