返回一个整数数组中最大子数组的和1

2016-04-06 17:32

一.设计思想

　　首先最大子数组maxsum和开始位置值maxstart都是数组第一个元素，然后判断开始位置的值，如果maxstart小于0，则另maxstart等于0，然后maxstart在自身基础上加上下一个数组元素，再判断maxstart与maxsum的大小，若maxstart大于maxsum，则maxsum的值为maxstart，maxstart的值不变；以此递归，直至数组最后一个元素，最后maxsum的值就是这个数组中最大子数组的和。

二.出现的问题

　　1.刚开始没有记录开始位置的值，只是判断下一个值的正负，正的加，负的跳过，导致子数组不是连续的；

　　2.换用了判断开始位置的值，正的则子数组和最大值再加上下一个数组元素，负的则子数组和最大值变成下一个数组元素的值，这次能解决刚开始的问题，但是出现了如果数组的最后一个元素为负的话，也会加到子数组和中，还有就是如果数组的元素全是负数，最后算出的结果就是最后一个元素。

三.解决方案

　　将最大子数组和与开始位置的值分别定义变量，让开始位置值参与循环，每次循环后更新最大子数组和的值，直到循环结束。

四.源代码

package 子数组;

import java.util.Scanner;

public class ArrayMax {

public static void main(String[] args){

int number,maxsum,maxstart;

int i;

Scanner in = new Scanner(System.in);

System.out.println("请输入数组的大小：");

number=in.nextInt();

int array[]=new int[number];

System.out.println("请输入数组的值：");

for(i=0;i<number;i++)

{

array[i]=in.nextInt();

}

maxsum=array[0];

maxstart=array[0];

for(i=1;i<number;i++)

{

if(maxstart<0)

{

maxstart=0;

}

maxstart+=array[i];

if(maxstart>maxsum)

maxsum=maxstart;

}

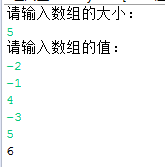
System.out.println(maxsum);

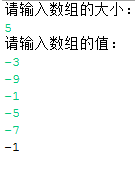
in.close();

}

}

五.结果截图

****

****

博客地址：http://www.cnblogs.com/java-meng/p/5360265.html