

区块链^Q文件转储系统

知识点栈链表 单调栈^Q 滑动窗

时间限制：2s 空间限制：256MB 限定语言：不限

题目描述：

区块链底层存储是一个链式文件系统，由顺序的 N 个文件组成，每个文件的大小不一，依次为 F_1, F_2, \dots, F_n 。随着时间的推移，所占存储会越来越大。

云平台考虑将区块链按文件转储到廉价的SATA^Q盘，只有连续的区块链文件才能转储到SATA盘上，且转储的文件之和不能超过SATA盘的容量。

假设每块SATA盘容量为 M ，求能转储的最大连续文件大小之和。

输入描述：

第一行为SATA盘容量 M ， $1000 \leq M \leq 1000000$

第二行为区块链文件大小序列 F_1, F_2, \dots, F_n 。其中 $1 \leq n \leq 100000$ ， $1 \leq F_i \leq 500$

输出描述：

求能转储的最大连续文件大小之和

示例1

输入:

1000
100 300 500 400 400 150 100

输出:

950

说明:

最大序列和为950, 序列为[400,400,150]

示例2

输入:

1000
100 500 400 150 500 100

输出:

1000

说明:

最大序列和为1000, 序列为[100,500,400]

解题思路：

使用滑窗来求得连续文件大小和的最大值。

如示例2：

left=0, count=100<1000, max=100

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----



left=0, right=1, count=100+300=400<1000, max=400, right+1=2

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----



left=0, right=2, count=400+500=900<1000, max=900, right+1=3

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----



left=0, right=3, count=900+400=1300>1000, max=900, left+1=1

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----



left=1, right=3, count=1300-100=1200>1000, max=900, left+1=2

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----



left=2, right=3, count=1200-300=900<1000, max=900, right+1=4

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----

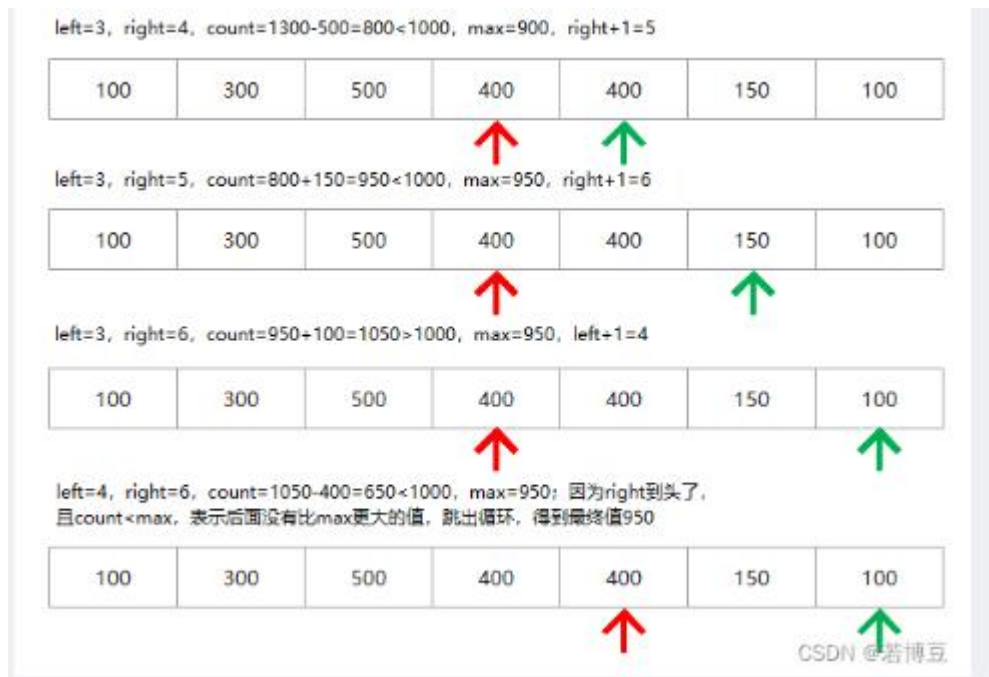


CSDN @若博豆

left=2, right=4, count=900+400=1300>1000, max=900, left+1=3

100	300	500	400	400	150	100
-----	-----	-----	-----	-----	-----	-----





```
public class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int M = sc.nextInt();
        sc.nextLine();
        String[] strings = sc.nextLine().split(" ");

        int left = 0;    //滑窗左边界
        int right = 1;   //滑窗右边界
        int count = Integer.valueOf(strings[0]);    //连续文件大小之和
        int max = 0;     //最大连续和
        while (max != M && (right < strings.length || count > max)){    //当 max==M 时直接图
退出, 当右边界到头且 count 小于 max 时退出
            if(count <= M){
                max = Math.max( max, count);
                if(right < strings.length){    //控制滑窗右边界
                    count += Integer.valueOf(strings[right]);
                    right ++;
                }
            }else {
                count -= Integer.valueOf(strings[left]);
                left ++;
            }
        }
    }
}
```

```
    }  
    System.out.println(max);  
}  
}
```