

# G Aftermath!!——刘京欣

## 题目描述

Aftermath——有劫后余生之意，在如今的网络文化中特指数学考试过后的惨状。

于是本题便是由此而来，劫后余生，出一道水题吧。

给你一个包含  $n$  个元素的整数序列，为  $a_1$ 、 $a_2$ 、 $\dots$ 、 $a_n$ 。

然后给你一个数  $i$ ，和一个数  $j$ ，求解上述序列中第  $i$  个元素与第  $j$  个元素间所有元素的和（第  $i$  个元素与第  $j$  个元素也包括在内）

## 输入

多组测试数据。

对于每组数据，第一行为两个整数  $n, m$  ( $1 \leq m, n \leq 500000$ )，用空格隔开，分别表示序列元素的个数  $n$  与查询的个数  $m$ ；

第二行为  $n$  个整数，用空格隔开，整数取值均在  $[0, 100]$  之间。

接下来  $m$  行，每行为两个整数  $i, j$  ( $1 \leq i \leq j \leq n$ )，具体含义见题目描述

## 输出

对于每组数据，输出  $m$  行，每行一个整数，对应一次查询的结果。

每组数据之间输出一个换行

## 输入样例

```
3 3
1 2 3
1 2
1 3
2 3
1 1
1
1 1
```

## 输出样例

```
3
6
5

1
```

## HINT

本题使用 `cin` 和 `cout` 会超时  
请使用 `scanf` 与 `printf`

## 解题思路

本题是这次上机题中 TLE 的人比较多的题目，主要在于循环的使用和 `scanf`, `printf` 与 `cin`, `cout` 的效率问题。大家在做的过程当中很容易想到嵌套多个循环，但这样即使用了 `cin`, `cout` 也同样会超时，虽然有时候超时只是 1, 2ms。因此我们需要做的是如何优化我们的算法，降低算法的时间复杂度，这样才不会导致超时。

这道题本身不难，是一道很基础的题目，但我们在做的过程中为了防止 TLE，可以在提交之前自己对程序进行判断。对于最大数据 500000，判断自己的程序需要算多少次。这是我在上机时渣诚学长给我说的方法，先膜拜一下我们的渣诚助教 Orz。后来我查找了部分资料，和大家分享一下。先给出已知，计算机每秒钟计算 10 亿次。而对于循环 `for(int i=0;i<n;i++)` 来说它的时间复杂度是  $O(n)$ ，表示进行  $n$  次计算；而两个 `for(int i=0;i<n;i++)` 循环嵌套，它的时间复杂度为  $O(n^2)$ ，表示进行  $n$  的平方次计算。其他更深的计算方法大家可以问度娘哟~

其次说下 `cin`, `cout` 与 `scanf`, `printf` 的效率问题，给一个简单解释（还有其他原因）：`cin` 和 `cout` 没有明确规定数据的输入和输出格式，系统需要检查数据类型后方可输出，而 `scanf`, `printf` 明确规定了数据格式（`%d` 等等），系统省略了判断数据类型这一环节，故速度会较快。所以大家要用好 `scanf("%d",&s)` 和 `printf("%d",s)`。

总之，优化算法，优化代码是我们努力的方向。

## 参考代码

```
#include<iostream>

#include <stdio.h>

int shu[500001];

//养成好习惯，数组较大时把它放在 main 函数外，防止占用过大内存(大家有兴趣可查一下栈溢出)

using namespace std;

int main()

{
```

```

int  n,m,i,j, num;

while (scanf("%d%d",&n,&m)!=EOF)//注意一定要写!=EOF, 否则会死循环
{
    shu[0]=0;

    for(int k=1;k<=n;k++)
    {
        scanf("%d",&num);

        shu[k] = shu[k-1] + num;

        //将数的和作为数组的元素而不是将输入的数作为数组的元素, 减少
        循环次数
    }

    for(int l=0;l<m;l++)
    {

        scanf("%d%d",&i,&j);

        printf("%d\n", shu[j]-shu[i-1]);

        //请大家回忆第四次上机的轮换镜台那道题, 都是从1开始算总数, 再将两个总
        数相减, 这个思想很好, 希望大家能够认真体会。同时注意在减的时候不要把
        i-1写成i 了。

    }

    printf("\n");

}

}

```

## 错误代码

(大家可以比较下两个代码的复杂程度和错误代码超时的原因)

```
#include<iostream>
```

```

#include<stdio.h>

using namespace std;

int shu[50001];

int main()
{
    int n,m,i,j,sum;

    while(scanf("%d%d",&n,&m)!=EOF)
    {
        for(int k=0;k<n;k++)
            scanf("%d",&shu[k]);

        for(int k=0;k<m;k++)
        {
            sum=0;

            scanf("%d%d",&i,&j);

            for(int s=i;s<=j;s++)
                sum+=shu[s-1];

            printf("%d\n",sum);
        }

        printf("\n");
    }
}

```

