

NNEZ_2019 级入门测试

时间：2019 年 8 月 4 日 8:30-11:30

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	序列	校霸	数数
英文题目与子目录名	sequence	nnez	count
可执行文件名	sequence	nnez	count
输入文件名	sequence.in	nnez.in	count.in
输出文件名	sequence.out	nnez.out	count.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	10
每个测试点分值	10	10	10
附加样例文件	无	无	无
结果比较方式	全文比较 (过滤行末空格及文末回车)		
题目类型	传统	传统	传统
运行内存上限	512M	64M	512M

二. 提交源程序文件名

对于 C++语言	sequence.cpp	nnez.cpp	count.cpp
对于 C 语言	sequence.c	nnez.c	count
对于 pascal 语言	sequence.pas	nnez.pas	count.pas

三. 编译命令 (不包含任何优化开关)

对于 C++语言	g++ -o sequence sequence.cpp -lm	g++ -o nnez nnez.cpp -lm	g++ -o count count.cpp -lm
对于 C 语言	gcc -o sequence sequence.c -lm	gcc -o nnez nnez.c -lm	gcc -o count count.c -lm
对于 pascal 语言	fpc sequence.pas	fpc nnez.pas	fpc count.pas

注意事项:

- 1、文件名 (程序名和输入输出文件名) 必须使用英文小写。
- 2、C/C++ 中函数 main() 的返回值类型必须是 int, 程序正常结束时的返回值必须是 0。
- 3、只提供 Linux 格式附加样例文件。
- 4、不允许使用万能头文件。

1,序列

(sequence.cpp/c/pas)

【问题描述】

LQ 要上大学了，她很开心。

看到高数课本，她很难过。

她想起了以前小学的日子，想起了平均数、极差、方差。于是她反手出了一道题：

对于给定序列 $\{a_n\}$ ，维护以下三个操作：

- 1, 输入 1 l r，输出区间[l,r]中数的极差
- 2, 输入 2 l r，输出区间[l,r]中数的平均值（保留两位小数）。
- 3, 输入 3 l r，输出区间[l,r]中数的方差（保留两位小数）

LQ 看了看你，便得意地走开了。

【输入格式】

输入文件名为 sequence.in

第一行两个正整数 n、m，n 表示序列中数的个数，m 表示询问次数。

第二行 n 个整数，表示给定的序列。

接下来 m 行，每行三个正整数 opt、l、r，其中 opt 表示对应的询问类型编号，l、r 代表该次询问即区间为[l,r]

若 opt=1，则答案为一个整数

若 opt=2 或 opt=3，则答案保留两位小数

【输出格式】

输出文件名为 sequence.out

输出 m 行，每行一个数，表示该次询问的答案。

【输入输出样例 1】

sequence1.in	sequence1.out
5 4	4.00
1 -3 6 -8 7	31.44
3 1 2	14
3 1 5	1.33
1 2 4	
2 1 3	

【输入输出样例 1 解释】

对于第一个操作，求区间[1,2]的方差。该区间平均数为-1，方差为
 $([1-(-1)]^2 + [(-3)-(-1)]^2)/2 = 4.00$

对于第二个操作，求区间[1,5]的方差。该区间的平均数为 0.6，方差为
 $[(1-0.6)^2 + [(-3)-0.6]^2 + (6-0.6)^2 + [(-8)-0.6]^2 + (7-0.6)^2]/5 = 31.44$

对于第三个操作，求区间[2,4]的极差。该区间的最大值为 6，最小值为(-8)，极差为 $6-(-8)=14$ 。

对于第四个操作，求区间[1,3]的平均值。平均值为 $[1+(-3)+6]/3=1.33$

【tips】

平均数: $\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n)$

方差: $s^2 = \frac{1}{n}[(x_1 - x)^2 + (x_2 - x)^2 + \cdots + (x_n - x)^2]$ (式中 x 为平均数)

极差: 最大值-最小值

【数据规模与约定】

本题共有 10 个测试点，每个测试点 10 分，共 100 分。

测试点编号	n	Q	opt=1	opt=2	opt=3
1	≤ 10	≤ 10	✓	✓	✓
2	≤ 100	≤ 100			
3		≤ 1	×	×	×
4			✓		
5	≤ 5000	≤ 5000	×	✓	✓
6			✓	×	×
7					
8			×	✓	✓
9	≤ 100000	≤ 100000	×		
10			✓		

其中×表示询问中无该项询问。

对于测试点 1-10: 保证 $1 \leq n, Q \leq 100000$ ，初始序列中的数 $-10^9 \leq a_i \leq 10^9$ 。

2,校霸

(nnez.cpp/c/pas)

【问题描述】

校霸今天心情大好，想要去教学楼巡视它的小弟们。

将 NNEZ 的校园看作一个无限大的网格图，教学楼是其中的一个矩形区域，其中每一个整点（横纵坐标均为整数的点）都是教室。

校霸有两种跳跃方式。校霸从未济湖 $(0,0)$ 出发，当采用跳跃方式 1 时会向右跳 x_0 个单位，向上跳 y_0 个单位。当采用跳跃方式 2 时会向右跳 x_1 个单位，向上跳 y_1 个单位。当然，校霸也可以采用方式 1 或方式 2 后退，即向左跳 x_0 (或 x_1) 个单位，向下跳 y_0 (或 y_1) 个单位。

问题来了，校霸能跳到教学楼中的多少个教室呢？

【输入格式】

输入文件名为 `nnez.in`

第一行四个正整数 U D L R 分别表示教学楼的上下左右边界（教学楼位于 $y=U$ $y=D$ $x=L$ $x=R$ 四条直线及它们所围成的矩形区域中）

第二行四个非负整数 x_0 y_0 x_1 y_1 ，其含义见问题描述

【输出格式】

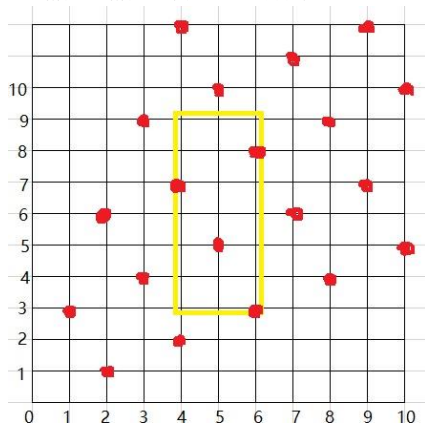
输出文件名为 `nnez.out`

输出一个整数，表示答案。

【输入输出样例 1】

nnez1.in	nnez1.out
9 3 4 6 1 3 2 1	4

【输入输出样例 1 解释】



黄色框为教学楼区域，红点为校霸可以跳到的点

【输入输出样例 2】

nnez2.in	nnez2.out
935 246 123 666 12 23 3 1	4365

【数据规模与约定】

本题共有 10 个测试点，每个测试点 10 分，共 100 分。

测试点编号	所有输入的数	特殊性质
1	≤ 10	$x_0=y_0=0$
2		
3	≤ 100	$L=R$
4		
5	≤ 1000	$x_0=x_1=0$
6		
7	≤ 10000	$x_0=y_1=0$
8		
9	≤ 100000	$L=R \ D=U$
10		

对于测试点 1-10: 保证 $1 \leq D \leq U \leq 10^5$, $1 \leq L \leq R \leq 10^5$, $0 \leq x_0, y_0, x_1, y_1 \leq 10^5$ 且这四个数不同时为 0.

3. 数数

(count.cpp/c/pas)

【问题背景】

LQ是个热爱学习的女孩子。

今天她学习了一维前缀和、一维打标记、二维前缀和。这是她的学习笔记。

一维前缀和：（以第一个元素开始的连续的一段数字的和，即解法中 S_i ）

（题目）给定一个有 n 个数的序列，其中第 i 个数为 a_i ， Q 次询问，每次询问输入一个正整数 x_i ，输出区间 $[1, x_i]$ 的数之和

（解法）设 S_i 为区间 $1 \sim i$ 的数和，则有递推式 $S_i = S_{i-1} + a_i$ 。即区间 $[1, i]$ 的数的和，等于区间 $[1, i-1]$ 的数的和，加上 a_i 。设第 t 次询问输入为 x ，输出 S_x 即可。

代码见选手目录下 example1.cpp

一维打标记：

（题目）给定一个有 n 个数的序列，其中第 i 个数为 a_i ，先有 Q 次修改操作，每次修改输入 3 个正整数 x_i, l_i, r_i ，表示对区间 $[l_i, r_i]$ 每个数加 x_i

在进行完 Q 次修改之后，有 M 次询问，每次输入一个正整数 x ，输出 Q 次修改后的 a_x 的值。

（解法）开一个辅助数组 t_i ，初始时元素都为 0。在第 i 次修改对区间 $[l_i, r_i]$ 每个数加 x_i 时，对 t_{l_i} 加上 x_i ， t_{r_i+1} 减去 x_i 。在 Q 次修改之后，设 t 数组的前缀和为 S_i 。求出前缀和数组 S ，则 Q 次修改之后第 p 个位置的数为 $a_p + S_p$ 。

（详细说明）设 $n=6, Q=1$ ，初始序列 a 为 1 2 3 4 5 6，修改操作为对 $[2, 4]$ 的每一个数加上 8

则初始的 t 序列为 0 0 0 0 0 0。按照解法描述， t_2 加上 8， t_{4+1} 即 t_5 减去 8。修改后 t 序列为 0 8 0 0 -8 0

对 t 序列求前缀和 S ：0 8 8 8 0 0。显然修改后的每个数为第 p 个位置的数为 $a_p + S_p$ 。

当 Q 不等于 1 时，该做法正确性显然。

代码见选手目录下 example2.cpp

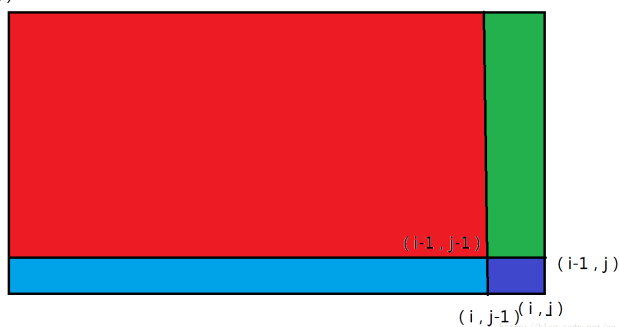
二维前缀和：（以位置 $(1, 1)$ 元素为左上角矩阵中数字之和，即解法中 $S_{i, j}$ ）

（题目）给定一个 $n*m$ 的矩阵 a ，其中第 i 行第 j 列元素为 $a_{i, j}$ 。 Q 次询问，每次给定输入给定 l_i, r_i 。求以位置 $(1, 1)$ 为左上角，位置 (l_i, r_i) 为右下角的矩阵区域所有元素之和。

（解法）设 $s_{i, j}$ 为以位置 $(1, 1)$ 为左上角，位置 (i, j) 为右下角的矩阵区域所有元素之和。则有递推式 $s_{i, j} = s_{i-1, j} + s_{i, j-1} - s_{i-1, j-1} + a_{i, j}$ 。对于询问 (l_i, r_i) ，输出 s_{l_i, r_i} 即可。

（详细说明） $s_{i, j}$ （图中所有区域）= $s_{i-1, j}$ （图中红+浅蓝）+ $s_{i, j-1}$ （图中红+深绿）- $s_{i-1, j-1}$ （图中红色区域，在前两次相加时被重复计算）+ $a_{i, j}$ （图中深蓝）

代码见选手目录下 example3.cpp



【问题描述】

LQ 想知道，是否存在二维打标记的做法？

即在一个给定的 $n*m$ 数字矩阵 a 中，有 Q 次修改操作，每次输入五个正整数 l_1, r_1, l_2, r_2, x ，表示对以 (l_1, r_1) 为左上角， (l_2, r_2) 为右下角的矩阵区域中所有数加上 x 。 Q 次操作之后，有 M 次询问。每次输入两个正整数 x, y ，输出在 Q 次修改之后 $a_{x,y}$ 的值。

【输入格式】

输入文件名为 `count.in`

第一行两个正整数 n, m ，表示这个矩阵有 n 行 m 列

接下来 n 行，每行 m 个整数，其中第 i 行第 j 个整数表示 $a_{i,j}$

接下来一行一个正整数 Q ，表示有 Q 次修改操作

接下来 Q 行，每行 5 个正整数 l_1, r_1, l_2, r_2, x ，其含义见问题描述。

接下来一行，一个正整数 M ，表示有 M 次询问。

接下来 M 行，每行两个正整数 x, y ，其含义见问题描述。

【输出格式】

输出文件名为 `count.out`

输出 m 行，每行一个正整数。其中第 i 行的正整数表示第 i 次询问的答案。

【输入输出样例 1】

count1.in	count1.out
3 4	3
5 8 3 4	4
3 4 5 8	5
5 8 3 4	8
2	7
1 1 2 2 3	
2 3 3 4 6	
5	
1 3	
1 4	
3 1	
3 2	
2 2	

【输入输出样例 1 解释】

初始：	5 8 3 4	第一次操作后：	8 11 3 4	第二次操作后：	8 11 3 4
	3 4 5 8		6 7 5 8		6 7 11 14
	5 8 3 4		5 8 3 4		5 8 9 10

此时 $(1, 3)$ 为 3， $(1, 4)$ 为 4， $(3, 1)$ 为 5， $(3, 2)$ 为 8， $(2, 2)$ 为 7

【输入输出样例 2】

见选手目录下 `count/count2.in` 和 `count/count2.out`

【数据规模与约定】

本题共有 10 个测试点，每个测试点 10 分，共 100 分。

测试点编号	n	Q, M	特殊性质
1	≤ 5	≤ 1	✕
2	≤ 100	≤ 100	
3			
4	≤ 1000	≤ 5000	✓
5			✕
6			
7	≤ 1000	≤ 100000	✓
8			✕
9			
10			

特殊性质 ✓: $l_1=1, r_1=1, l_2=n, r_2=m$

对于测试点 1-10: 保证 $1 \leq n \leq 1000, 1 \leq Q \leq 100000, -10^9 \leq a_{i,j} \leq 10^9$