题目翻译

1. 星座3

内存：537MB 时限：1s 题型：传统题

/usr/bin/g++ -DEVAL -std=gnu++14 -O2 -pipe -static -s -o constellation3 constellation3.cpp

题目描述：

JOI-kun 拍了一张夜景照片。照片由N×N个像素组成，即沿水平和垂直方向的N个像素。从左开始的第x列和从下开始的第y行（1≤x≤N，1≤y≤N）中的像素称为像素（x，y）。

图片中的每个像素都显示建筑物、夜空或星星。它们的颜色相应地分别是白色、黑色或黄色。

对于1≤i≤N的每个i，在第i列中，从最下面一行到从下往上数第Ai行是表示建筑物的白色像素。有M个黄色像素显示星星。第j个黄色像素（1≤j≤M）是像素（Xj，Yj）。所有其他像素都是表示夜空的黑色像素。

如果满足图中的矩形区域满足以下两个条件，则表示一个星座：

• 矩形区域中没有白色像素

• 矩形区域中有两个或以上黄色像素

JOI-kun已经厌倦了看星座。他想通过把一些黄色像素画成黑色使得图片上没有矩形区域表示星座。然而，如果他绘制太多黄色像素会让图片变得不自然。更准确地说，如果他把第j个黄色像素（1≤j≤M）涂成黑色，那么图像的不自然程度将增加Cj. 最初，图片的不自然程度为0。

编写一个程序，给定图片信息和每个黄色像素的将其绘制为黑色将增加的不自然程度，计算在画出一些黄色像素后，图片没有矩形区域显示星座的最小不自然程度。

题目输入：

所有输入为整数

N

A1 ... AN

M

X1 Y1 C1

...

XM YM CM

题目输出：

输出一行，表示将一些黄色像素涂黑使得没有矩形区域显示星座后，图片的最小非自然水平。

数据限制：

• 1 ≤ N ≤ 200 000.

• 1 ≤ Ai ≤ N (1 ≤ i ≤ N).

• 1 ≤ M ≤ 200 000.

• 1 ≤ Xj ≤ N (1 ≤ j ≤ M).

• 1 ≤ Yj ≤ N (1 ≤ j ≤ M).

• 1 ≤ Cj ≤ 1 000 000 000 (1 ≤ j ≤ M).

• AXj < Yj (1 ≤ j ≤ M).

• (Xj, Yj) , (Xk, Yk) (1 ≤ j < k ≤ M).

子任务：

1. (14 points) N ≤ 300, M ≤ 300.

2. (21 points) N ≤ 2 000, M ≤ 2 000.

3. (65 points) No additional constraints.

样例一：

输入：

5

1 3 4 2 3

3

1 5 3

4 3 2

2 4 2

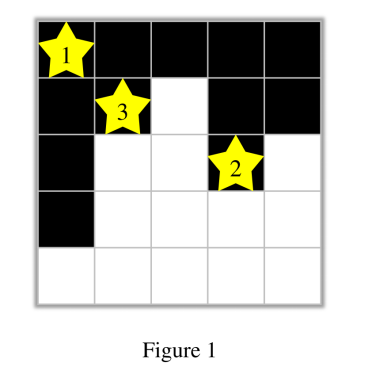
输出：

2

解释：

左上角为 (1, 5)，右下角为 (2, 4) 的矩形区域表示了一个星座. 如果 JOI-kun 涂黑第3个黄色像素，不自然程度将增加 2 并且图中不会再有星座，因此最小值为2.

下图为样例一的输入



样例二：

输入：

7

5 6 2 3 6 7 6

5

7 7 5

3 3 7

3 7 10

1 7 6

4 7 8

输出：

16

解释：

最优解涂黑第3个和第4个黄色像素

样例三：

输入：

8

6 8 5 7 3 4 2 1

10

8 2 9

6 6 7

8 3 18

5 8 17

8 5 3

5 5 3

5 4 8

1 8 13

1 7 5

7 4 13

输出：

44

1. 收获

内存：537MB 时限：3s 题型：传统题

/usr/bin/g++ -DEVAL -std=gnu++14 -O2 -pipe -static -s -o harvest harvest.cpp

题目描述：

IOI农场是一个种植苹果的农业农场。它以坐落在一个大的环形湖周围而闻名。

IOI农场有N名员工，从1到N，有M棵苹果树，从1到M。这个湖的周长是L米。

一开始，员工i（1≤i≤N）在距离湖最北端点顺时针方向Ai米处等待。Ai（1≤i≤N）的值是不同的。苹果树j（1≤j≤M）生长在距湖最北端顺时针方向Bj米处。Bj（1≤j≤M）的值是不同的。此外，任何员工的初始位置都没有苹果树。

由于IOI农场苹果树的特殊品种改良，每棵苹果树最多只能同时结至多一个苹果。而且，如果一个苹果被从苹果树上摘下来，它就会有一个新的苹果在恰好C秒后长出。在时间0，每棵苹果树上都有一个苹果，并且每个员工开始在湖边沿着顺时针方向散步。每个员工的速度是每秒1米。如果员工到达苹果树时树上有一个苹果，那么员工总是会收获它（如果一棵苹果树员工到达的时间有一个新苹果恰好长出，员工也会收获）。我们忽略员工收获苹果的时间。

K总统是IOI农场的股东。既然你是IOI农场的经理，K总统让你汇报员工的工作效率。更准确地说，K总统想知道以下Q个值。

对于每个k（1≤k≤Q），员工Vk在Tk之前收获的苹果数量（包括可能有的正好在Tk时间收获的苹果）。

写一个程序，给定雇员的数量，苹果树的数量，湖水的周长，一棵苹果树长出一个新苹果所需的时间，员工和苹果树的位置和Q个询问的信息，计算每个询问收获苹果的数量。

题目输入：

所有输入均为整数

N M L C

A1 ... AN

B1 ... BM

Q

V1 T1

...

VQ TQ

题目输出：

输出Q行，第k行表示第k个询问的答案

数据限制：

• 1 ≤ N ≤ 100 000.

• 1 ≤ M ≤ 100 000.

• N + M ≤ L ≤ 1 000 000 000.

• 1 ≤ C ≤ 1 000 000 000.

• 0 ≤ Ai < L (1 ≤ i ≤ N).

• Ai < Ai+1 (1 ≤ i ≤ N-1).

• 0 ≤ Bj < L (1 ≤ j ≤ M).

• Bj < Bj+1 (1 ≤ j ≤ M 1).

• Ai , Bj (1 ≤ i ≤ N, 1 ≤ j ≤ M).

• 1 ≤ Q ≤ 100 000.

• 1 ≤ Vk ≤ N (1 ≤ k ≤ Q).

• 1 ≤ Tk ≤ 1 000 000 000 000 000 000 = 10^18 (1 ≤ k ≤ Q).

子任务：

1. (5 points) N ≤ 3 000, M ≤ 3 000, Q ≤ 3 000.

2. (20 points) Tk ≧ 1 000 000 000 000 000 = 10^15 (1 ≦ k ≦ Q)．

3. (75 points) No additional constraints.

样例一：

输入：

3 2 7 3

1 4 6

0 5

3

1 7

2 3

3 8

输出：

2

1

1

解释：

• 时间1时，雇员 2 收获树 2的苹果, 雇员 3 收获树 1的苹果

• 时间3时, 雇员 2 到达树1，树1此时无苹果，所以不收获

• 时间4时, 雇员 1收获树 2的苹果

• 时间6时， 雇员 1 收获树 1的苹果。雇员 3到达树2，但是无苹果收获

• 时间8时, 雇员 2 收获树 2的苹果，雇员 3到达树1，但是树1无苹果

由于员工1在时间7之前收获了2个苹果，所以第一行输出2。

样例二：

输入：

5 3 20 6

0 4 8 12 16

2 11 14

9

4 1932

2 93787

1 89

5 98124798

1 2684

1 137598

3 2

3 8375

4 237

输出：

146

7035

7

7359360

202

10320

0

628

18

样例三：

输入：

8 15 217 33608

0 12 71 96 111 128 152 206

4 34 42 67 76 81 85 104 110 117 122 148 166 170 212

14

2 223544052420046341

3 86357593875941375

4 892813012303440034

1 517156961659770735

7 415536186438473633

6 322175014520330760

7 557706040951533058

6 640041274241532527

5 286263974600593111

8 349405886653104871

1 987277313830536091

5 989137777159975413

2 50689028127994215

7 445686748471896881

输出：

33230868503053

3

5

1

123542793648997

8

165811220737767

8

7

1

1

7

7535161012043

132506837660717

1. 流浪猫

内存：537MB 时限：2s 类型：交互题

/usr/bin/g++ -DEVAL -std=gnu++14 -O2 -pipe -static -s -o Anthony\_Catherine stub.cpp Anthony.cpp Catherine.cpp

题目描述：

Anthony是一只生活在JOI城的蚂蚁。JOI城有N个小镇，从0到N-1。Anthony生活在0号镇。共有M条道路，编号从0到M-1。道路i（0≤i≤M-1）连接城镇Ui和镇Vi，并且可以从两个方向通过。不同的道路连接着不同的城镇对。从任何一个城镇经过若干条路后都可以到其他任何一个城镇。

Anthony的朋友Catherine是一只猫。她计划去参观JOI城，但她不知道路的信息并且她经常迷路。Anthony决定提前在路上做标记。有A种标记类型，从0到A-1编号。

现在Catherine来到了JOI城的一个小镇。当她在小镇0以外的城镇时，她会做出如下举动：

对于每种类型的标记，她都可以计算出从她现在所在的城镇出发的那类标记的道路的数量，除了她最后经过的那条路（如果有这条路的话）。

之后，她选择一条路出发。请注意，除了她最后经过的那条路外，她只能通过路的标记来区分路。通过选择合适的道路，她想在不花费太多的时间下到达小镇0。更准确地说，若从她的第一个城镇到镇0经过道路数量的最小值是d，她想通过选择最多d+B次的道路到达镇0。

编写一个程序，根据道路信息，实现Anthony放置标记的策略，以及Catherine选择道路的策略。

实现细节：

你需要提交两个文件

第一个文件是Anthony.cpp。它负责实现Anthony的策略，具体来说包括如下的函数。这个程序需要包含头文件Anthony.h。

• std::vector<int> Mark(int N, int M, int A, int B,

std::vector<int> U, std::vector<int> V)

这个函数只会在程序开始时调用一次。

◦ 参数N是小镇的数量N.

◦ 参数M是道路的数量M

◦ 参数A是标记的种类数A

◦ 参数B是Catherine 选择道路的冗余次数B。

◦ 参数U和V是长度为M的数组，U[i]和V[i]代表道路i连接的小镇Ui和Vi(0 ≤ i ≤ M-1).

◦ 返回值x应该是长度为M的数组。如果数组长度不为M，程序会提示错误信息 Wrong Answer [1]. x[i]的值(0 ≤ i ≤ M-1) 代表道路i的标记. 要求0 ≤ x[i] ≤ A-1 。如果x[i]的值不满足该范围，则程序会提示错误信息Wrong Answer [2].

第二个文件是Catherine.cpp。它负责实现Catherine的策略，具体来说包括如下的函数。这个程序需要包含头文件Catherine.h。

• void Init(int A, int B)

这个函数只会在程序开始时被调用一次

◦ 参数A是标记的种类数A

◦ 参数B是Catherine 选择道路的冗余次数B。

• int Move(std::vector<int> y)

这个函数在任何Catherine 到达一个非0号城镇时都会被调用

◦ 参数y是一个长度为A的数组，y[j]表示从当前所处的城镇出发标记种类为j的且不是她上一次通过的道路（如果该道路存在）的数量 (0 ≤ j ≤ A-1).

◦ 返回值z应该满足1 ≤ z ≤ A-1. 如果不满足，程序会提示错误信息Wrong Answer [3]. 如果z = -1, 她选择回头走她刚才通过的道路. 如果 0 ≤ z ≤ A-1, 她选择一条标记为z的道路。当这个函数第一次被调用时，如果 z = -1, 程序会提示错误信息Wrong Answer [4]. 如果 0 ≤ z ≤ A-1 并且 y[z] = 0, 你的程序会提示错误信息 Wrong Answer [5].

如果Catherine选择了一条与她上一次走过的路不同的路，她实际上会走的路的标记将由函数Move的返回值指定，并且选择这个标记中的路的其中一条。注意这个选择不能随机。

如果Catherine在经过d+B次道路后没有到达城镇0（即，如果已经调用Move函数 d+B次），你的程序被判定为Wrong Answer[6]。

重要提示：

• 您的程序可以实现其他内部使用的函数，或使用全局变量。提交的文件将与评分器一起编译，并成为单个可执行文件。所有全局变量和内部变量函数应在未命名的命名空间中声明，以避免与其他文件冲突。评分时，它将被执行为Anthony和Catherine两个进程。Anthony的进程和Catherine的进程不能共享全局变量。

• 程序不得使用标准输入和标准输出。程序不能用任何方法与其他文件通信。但是，程序可能会将调试信息输出到标准错误流中。

编译运行：

您可以从竞赛网页下载一个存档文件，其中包含样例评分程序来测试你的程序。存档文件还包含你的程序的示例源文件。

样例评分程序是文件grader.cpp。为了测试你的程序，把grader.cpp，Anthony.cpp，

Catherine.cpp、Anthony.h、Catherine.h放在同一目录中的，并运行以下命令编译

你的程序。

g++ -std=gnu++14 -O2 -o grader grader.cpp Anthony.cpp Catherine.cpp

编译成功后，将生成可执行文件grader。

注意，实际的评分程序与样本评分程序不同。样本评分程序将作为单个进程，它将从标准输入读取输入数据，并将结果写入标准输出。

样例评分程序的输入：

样本评分程序将从标准输入流中读取以下数据

N M A B S

U0 V0

...

U{M-1} V{M-1}

S为Catherine起始城镇的编号

样例评分程序的输出：

当程序成功终止时，样例评分程序将以下信息写入标准输出（为清晰起见用引号标注出来）：

•如果你的程序提示了错误信息Wrong Answer [1], [2], [3], [4] ,[5],它会输出错误信息“Wrong Answer [1]”.

• 如果Catherine 经过N+B次移动后没有到达城镇0, 它会输出错误信息“Wrong Answer; Number of moves > N + B”.

• 否则它会输出移动的次数(函数Move调用的次数)，如“Number of moves = 4”. 注意它不会判断移动是正确的还是Wrong Answer [6].

如果您的程序被判断为几种类型的错误答案，则示例分级器只报告其中一种。

在样例评分程序机中，当Catherine 选择一条与她上一次通过的道路不同的道路时，她实际要经过的道路是在函数Move返回的标记值的道路中随机选择的，使用具有固定种子的伪随机数。如果要使用不同的种子，执行分级机如下：

./grader 2020

这里的第一个参数是一个整数，它是伪随机数的种子。

数据限制：

• 2 ≤ N ≤ 20 000.

• 1 ≤ M ≤ 20 000.

• 1 ≤ S ≤ N-1 (S是 Catherine出发城镇的编号).

• 0 ≤ Ui < Vi ≤ N-1 (0 ≤ i ≤ M-1).

• (Ui, Vi) , (Uj, Vj) (0 ≤ i < j ≤ M-1).

• 有可能可以从一个城镇经过多条道路到达另一个城镇

子任务：

1. (2 points) A = 4, B = 0, M = N-1.

2. (2 points) A = 4, B = 0.

3. (2 points) A = 3, B = 0, M = N-1.

4. (9 points) A = 3, B = 0.

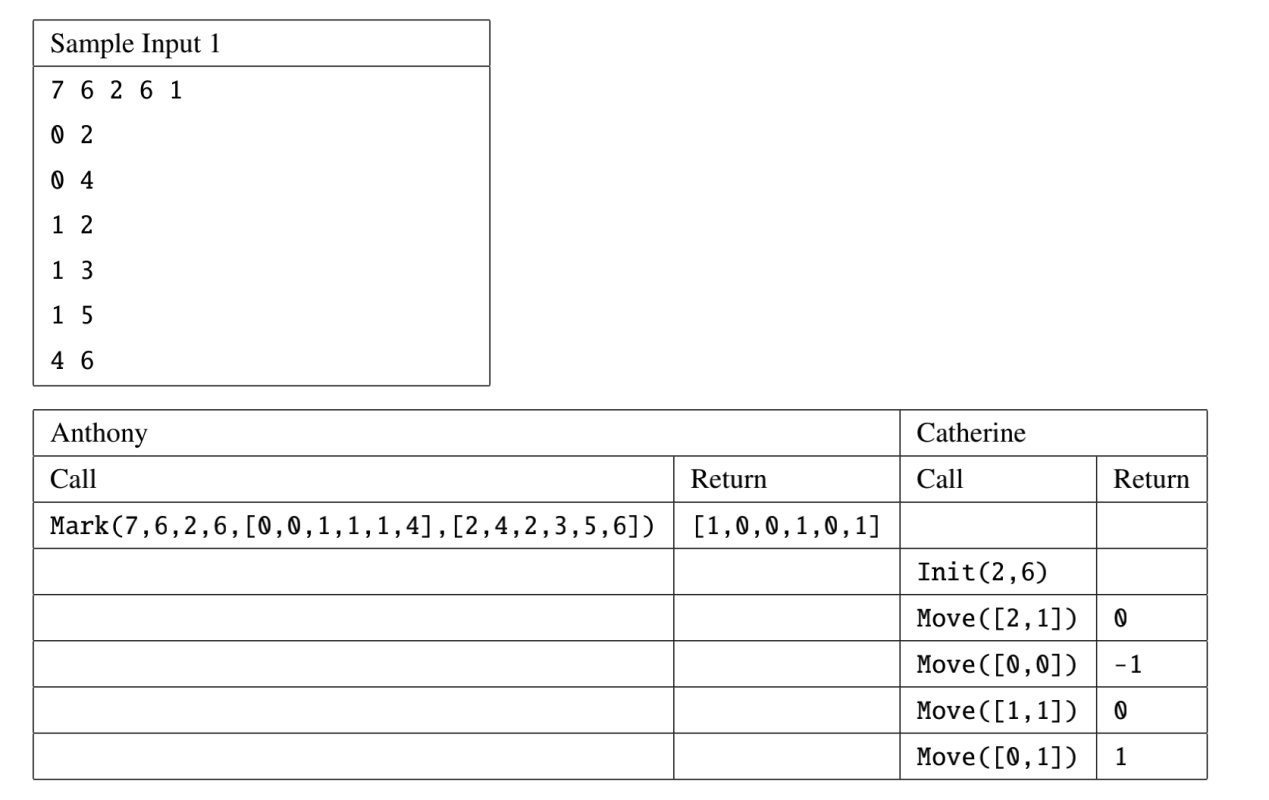
5. (5 points) A = 2, B = 2N, M = N-1, 6 ≤ N ≤ 500.

6. (71 points) A = 2, B = 12, M = N-1.

7. (9 points) A = 2, B = 6, M = N-1.

样例通信过程

以下是一个样例评分程序的样例输入和对应的函数调用过程

在这个样例通信过程中，Catherine按顺序访问了城镇 1, 5, 1, 2, 0. 我们已知d = 2. Catherine

移动了4次。

这个样例输入满足子任务7的限制.

在竞赛网站上可以下载的文件中, sample-02.txt 满足子任务4的限制.