

Public NOIP Round #4

普及组

时间：2022 年 11 月 20 日 8:30 ~ 12:00

题目名称	咖啡	画图	治病	拓扑序计数
题目类型	传统型	传统型	传统型	传统型
可执行文件名	coffee	paint	doctor	topo
每个测试点时限	1.0 秒	1.0 秒	3.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	10	4	5	13
测试点是否等分	是	否	是	否

提交源程序文件名

对于 C++ 语言	coffee.cpp	paint.cpp	doctor.cpp	topo.cpp
-----------	------------	-----------	------------	----------

编译选项

对于 C++ 语言	-lm -O2
-----------	---------

注意事项：

1. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
2. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
7. 评测时采用的机器配置为 Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz (QOJ 与 Public Judge)。使用的操作系统为 Ubuntu 18.04，编译器版本为 g++ 11.1.0 (Ubuntu 11.1.0-1ubuntu1 18.04.1)，上述时限以此为准。
8. **特别注意，在提交时不需要使用文件输入输出。选手应在标准输入中读入数据，并将答案输出至标准输出。**

咖啡 (coffee)

【题目描述】

在花花大学,学生需要按照顺序去 n 间教室上课。第 i 间教室有一个属性 $s_i \in \{0, 1\}$ 。若 $s_i = 1$, 表示这间教室有一座咖啡机; 若 $s_i = 0$, 表示这间教室没有咖啡机。

你是花花大学的一名学生。在一间有咖啡机的教室里,你可以通过饮用一杯咖啡来使得自己不会犯困。特别地,在你离开一间有咖啡机的教室后,你可以携带最多两杯咖啡(每只手可以拿一杯)前往下一间教室,这样即使那间教室没有咖啡机,你可以通过引用你携带的咖啡来提神。

现在你要知道,你最多可以在多少间教室内饮用咖啡。

【输入格式】

输入的第一行包含一个整数 n 。

接下来一行,包含一个长度为 n 的,只由字符 '0' 与字符 '1' 构成的字符串 s , 其中第 i 个字符 s_i 描述了第 i 间教室是否有咖啡机。

【输出格式】

输出一行一个整数,表示答案。

【样例 1 输入】

```
1 6
2 010100
```

【样例 1 输出】

```
1 5
```

【样例 2 输入】

```
1 10
2 0000000110
```

【样例 2 输出】

1 3

【样例 2 解释】

369 是 3 的倍数。

【样例 3】

见选手目录下 *coffee/coffee3.in* 与 *coffee/coffee3.ans*。样例 3 满足子任务 3 的性质。

【子任务】

对于 100% 的数据, $1 \leq n \leq 10^5$ 。

测试点编号	$n \leq$
1	1
2 ~ 6	10
7 ~ 8	5 000
9 ~ 20	10^5

画图 (paint)

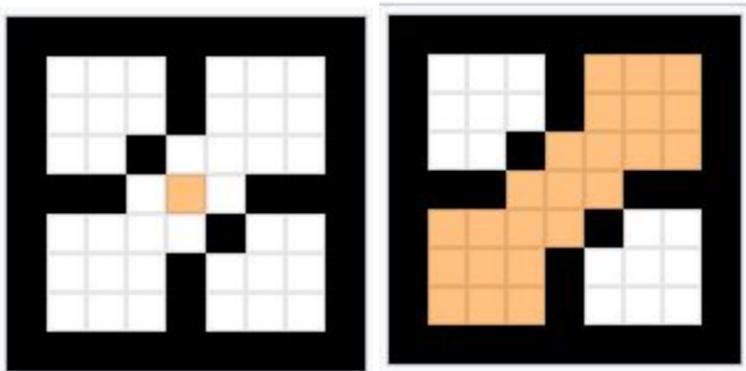
【题目描述】

在画图软件中，你最喜欢的便是油漆桶工具。

我们将画图软件中的画布视为一个 $N \times M$ 的由像素构成的矩阵。第 i 行第 j 列的像素记为 $R_{i,j}$ ($0 \leq i < N$, $0 \leq j < M$ ，即行与列均从 0 开始编号)。由于你正在使用的电脑的机能有限，因此每个像素只能是 $0 \sim 9$ 以内的整数，共 10 种。

当我们在位置 (i, j) 使用油漆桶工具，将其涂成颜色 c 时，所有与 (i, j) 联通且颜色相同的像素的颜色都将会被更改为颜色 c 。

我们称两个像素 $A(x_a, y_a)$ 与 $B(x_b, y_b)$ 联通，当且仅当 $|x_a - x_b| + |y_a - y_b| = 1$ 且 A, B 的颜色相同，或存在另一个像素 C ，使得像素 A, C 之间联通，且像素 B, C 之间联通。即，两个像素联通，当且仅当存在一条路径，满足路径上任意两个相邻的像素四联通，且所有经过的像素的颜色均相同。



例如，在上图中对橙色位置使用油漆桶进行涂色，会将所有右图中对应的位置全部涂色。

现在，给定当前局面下的画布，你要求出，在位置 (x, y) 上使用颜色 c 进行油漆桶操作后画布的形态。

【输入格式】

输入的第一行包含两个整数 N, M 。

接下来 N 行，每行 M 个字符，描述像素构成的矩阵。每个字符均为 $0 \sim 9$ 中的整数。

最后一行包含三个整数 x, y, c ，表示进行的操作。

【输出格式】

输出 N 行，每行 M 个字符，表示操作后矩阵的形态。

【样例 1 输入】

```
1 6 7
2 0000000
3 0112211
4 0011210
5 0012210
6 0111110
7 1765671
8 1 2 3
```

【样例 1 输出】

```
1 0000000
2 0332233
3 0033230
4 0032230
5 0333330
6 1765671
```

【样例 2 输入】

```
1 1 1
2 0
3 0 0 0
```

【样例 2 输出】

```
1 0
```

【样例 3】

见选手目录下 *paint/paint3.in* 与 *paint/paint3.ans*。

【子任务】

对于所有数据, $1 \leq N, M \leq 1000$, $0 \leq x < N$, $0 \leq y < M$, $0 \leq c \leq 9$ 。

测试点编号	$N \leq$	$M \leq$	特殊性质
1 ~ 2	2	2	五
3 ~ 5	10	10	
6 ~ 7	1000	1000	$N = 1$ 或 $M = 1$
8 ~ 13	110	110	无
14 ~ 20	1000	1000	

治病 (doctor)

【题目描述】

虱子国王尼特这天有点不舒服，它周围的 n 个医生立刻开出了药方：第 i 个医生告诉它，从这天起的第 L_i 天到第 R_i 天，它应该服用 $x_{i,1}, x_{i,2}, \dots, x_{i,K_i}$ 这 K_i 种药，每天每种药应当服用恰好一片。注意，如果有多个医生的药方里都要求尼特在第 p 天服用第 q 种药，那尼特在第 p 天仍然只会服用一片第 q 种药。编号为 j 的药每片需要 c_j 元钱。

然而，由于尼特的疏忽，有恰好一位庸医混进了医生队伍里，但尼特并不知道哪位医生是庸医。所以它想知道，对于所有 $1 \leq i \leq n$ ，如果它按照除了第 i 个医生之外的所有医生的药方吃药，它总共将花费多少钱。

【输入格式】

第一行两个正整数 n, m ，分别为医生数和药片种类数。

接下来一行 m 个正整数 $c_1 \sim c_m$ 。

接下来 n 行，第 i 行描述第 i 个医生。首先三个正整数 L_i, R_i, K_i ，后面 K_i 个正整数 $x_{i,1}, x_{i,2}, \dots, x_{i,K_i}$ 。保证 $x_{i,1}, x_{i,2}, \dots, x_{i,K_i}$ 互不相同。

【输出格式】

输出 n 个非负整数，第 i 个表示，如果尼特认为第 i 个医生是庸医并除开他的药方，它总共将花费多少钱。

【样例 1 输入】

```

1 5 4
2 10000 1000 100 10
3 3 4 2 2 3
4 4 8 3 1 2 4
5 6 7 2 3 4
6 8 9 2 1 4
7 2 6 3 1 2 3

```

【样例 1 输出】

```

1 87660 75640 87560 77650 66460

```

【样例 1 解释】

这里仅解释输出中的第一个和第五个数。

如果第一位医生是庸医，则尼特：

- 在第 2,3 天会吃药片 1,2,3。花费 $11100 \times 2 = 22200$ 元。
- 在第 4,5,6,7 天会吃药片 1,2,3,4。花费 $11110 \times 4 = 44440$ 元。
- 在第 8 天会吃药片 1,2,4。花费 11010 元。
- 在第 9 天会吃药片 1,4。花费 10010 元。

总花费 87660 元。

如果第五位医生是庸医，则尼特：

- 在第 3 天会吃药片 2,3。花费 1100 元。
- 在第 4 天会吃药片 1,2,3,4。花费 11110 元。
- 在第 5 天会吃药片 1,2,4。花费 11010 元。
- 在第 6,7 天会吃药片 1,2,3,4。花费 $11110 \times 2 = 22220$ 元。
- 在第 8 天会吃药片 1,2,4。花费 11010 元。
- 在第 9 天会吃药片 1,4。花费 10010 元。

总花费 66460 元。

【样例 2】

见选手目录下 *doctor/doctor2.in* 与 *doctor/doctor2.ans*。样例 2 满足子任务 1 的性质。

【样例 3】

见选手目录下 *doctor/doctor3.in* 与 *doctor/doctor3.ans*。样例 3 满足子任务 3 的性质。

【样例 4】

见选手目录下 *doctor/doctor4.in* 与 *doctor/doctor4.ans*。样例 4 满足子任务 5 的性质。

【子任务】

本题捆绑测试，你需要通过一个子任务的所有测试点才能得到子任务的分数。

对于所有数据： $1 \leq n, m \leq 5 \times 10^5, 1 \leq L_i \leq R_i \leq 10^6, 1 \leq K_i \leq m, \sum K_i \leq 10^6$ 。

子任务编号	特殊性质	分数
1	$n \leq 100, m \leq 100, R_i \leq 100, \sum K_i \leq 100$	20
2	$n \leq 5000, m \leq 5000, R_i \leq 5000, \sum K_i \leq 10^4$	20
3	$[L_i, R_i]$ 互不相交	20
4	$m = 1$	20
5	无	20

拓扑序计数 (topo)

【题目描述】

本题中涉及到的图论定义：

- 一个 n 个点，点的编号为 $1, 2, \dots, n$ 的有向图 $G = (V, E)$ 的**拓扑序**是一个 $1, 2, \dots, n$ 的排列 p ，且若 E 中存在 $x \rightarrow y$ 的边，就有 p 中 x 出现在 y 之前。

今天，算法竞赛机器人小 G 学习了拓扑排序相关知识。凭着强大的机器学习本领，它很快便一并学会了如何计算一个有向无环图的拓扑序个数。接着，它开始思考一个拓展问题：给定一个有向无环图 G 和两个 G 中的点 u, v ，请你求出有多少种 G 的拓扑序满足 u 排在 v 之前。

你知道稍加思考后小 G 也能秒掉这题。不巧，就在这时候停电了，依靠插头进食的小 G 也因此停止工作了。所以你只好自己解决这个拓展问题了。

为了让问题更富有挑战性，设 G 中总点数为 n ，请你对所有 $n(n-1)$ 对 (u, v) 都求出答案。

【输入格式】

本题有多组数据，第一行是数据组数 T 。

对于每组数据：第一行两个正整数 n, m ，分别为 G 的点数和边数。接下来 m 行，每行两个正整数 x, y ，表示有向图里一条 $x \rightarrow y$ 的边。保证没有重边且 $x < y$ （也就是 $[1, 2, \dots, n]$ 总是一个合法拓扑序）。

保证同一个测试点中至多有 5 组数据满足 $n > 10$ 。

【输出格式】

对每组数据输出一个 $n \times n$ 的矩阵，第 i 行第 j 列是 $v = i, u = j$ 时的答案，注意 (v, u) 的顺序和 (i, j) 是反的。特别地，当 $i = j$ 时请你输出 0。

【样例 1 输入】

```
1 2
2 3 2
3 1 2
4 1 3
5 4 2
6 1 2
7 3 4
```

【样例 1 输出】

```
1 0 0 0
2 2 0 1
3 2 1 0
4 0 0 3 1
5 6 0 5 3
6 3 1 0 0
7 5 3 6 0
```

【样例 1 解释】

对于第一组数据，原图共有两种拓扑序 $[1, 2, 3], [1, 3, 2]$ 。满足 1 在 2 前面的有 2 种，所以答案矩阵的第 2 行第 1 列是 2；满足 3 在 2 前面的有 1 种，所以答案矩阵的第 2 行第 3 列是 1。

【样例 2】

见选手目录下 *topo/topo2.in* 与 *topo/topo2.ans*。样例 2 满足子任务 1 的性质。

【样例 3】

见选手目录下 *topo/topo3.in* 与 *topo/topo3.ans*。样例 3 满足子任务 10 的性质。

【子任务】

本题捆绑测试，你需要通过一个子任务的所有测试点才能得到子任务的分数。

对于所有数据： $1 \leq T \leq 100, 1 \leq n \leq 20, 0 \leq m \leq \binom{n}{2}$ ，**保证同一个测试点中至多有 5 组数据满足 $n > 10$ 。**

子任务编号	$n \leq$	$m \leq$	$T \leq$	分值
1	5	$\binom{n}{2}$	20	10
2	20	0		5
3		1		5
4		2		5
5		10		10
6	10	$\binom{n}{2}$	30	5
7	12		40	5
8	14		50	10
9	16		60	5
10	17		70	5
11	18		80	10
12	19		90	5
13	20		100	20