Beijing Team Selection Contest 1

January 27, 2018

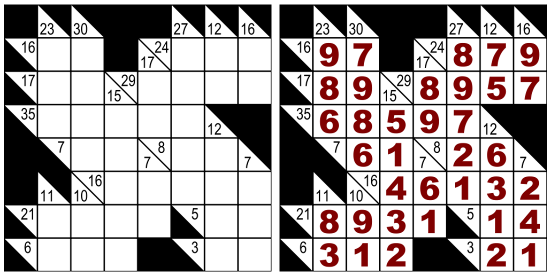
竞赛时间：4.5 hours

题目名称	Kakuro	Cross sums	Kreuzsummen
源程序名	kakuro	cross	kreuz
题目类型	传统型	传统型	传统型
输入文件名	kakuro.in	cross.in	kreuz.in
输出文件名	kakuro.out	cross.out	kreuz.out
单测试点时限	1s	2s	2s
空间限制	512MB	512MB	1024MB
测试点数目	20	20	20
单测试点分值	5	5	5
是否有 SPJ	否	是	否

首先介绍一下 Kakuro(カックロ) 这个游戏。

游戏规则为：

- 方形空格中填入 1 ∼ 9 的整数。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之和，左下角的数字等于其下方邻接之连续方格中数字之和。
- 无论是横向还是纵向，连续方格中的数字不能重复。



左边为一个 Kakuro 游戏，右边为这个游戏的唯一解。

我们称一开始给出的数字为**线索**，称需要填入数字的地方为**空格**。如果一个格子包含线索那么就不需要填入数字。我们约定所有的谜题都非空，即至少有一个空格需要被填入。

注意：在以下题目中的游戏规则可能会有所不同，请认真阅读在每个题目下的规则。

注意事项

- 比赛与评测均在 Windows 下。评测的时候使用 lemon 评测，请大家在选手目录下直接放三个文件，不需要子文件夹。
- 在原则上，编译会开 O2,C++11 和放大栈空间限制。具体会在考试开始后通知大家。

Problem A. Kakuro

输入文件: `kakuro.in`
输出文件: `kakuro.out`
时间限制: 1 second
空间限制: 512 megabytes

游戏规则：

- 空格中填入**正整数**。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之和，左下角的数字等于其下方邻接之连续方格中数字之和。

Apia 给了 Rimbaud 一个 Kakuro 谜题。心不灵手不巧的 Rimbaud 根本不会做 Kakuro，所以只在空格里面填上了一些随机的数字，称这个为一个局面，即包含了谜题一开始给出的线索和后面填入的数字。

现在 Rimbaud 希望能修改这个局面使得她的答案是一个合法解。这个局面中有些数字 (**包括一开始的给出线索和后面填入的数字**) 是可以修改的。每个数字都有个特定的代价，将这个数字加 1 或者减 1 都得付出这个数字对应的代价。注意对于一组合法解，必须满足游戏规则，也就是**空格中填的数字必须是正整数并且满足和的条件，但是不要求不重复**。

Rimbaud 希望用最少的代价让这个局面变得合法，如果不可能那么输出 -1。

Input

第一行，两个正整数表示 n, m 表示这个游戏的行和列。

接下来 n 行，每行包含 m 个 0 到 4 的数字，第 i 行第 j 列表示第 i 行第 j 列格子的种类。

- 0 表示这个格子既不是空格也不是线索。
- 1 表示这个格子左下角包含线索，右上角没有线索。
- 2 表示这个格子右上角包含线索，左下角没有线索。
- 3 表示这个格子左下角右上角都包含线索。
- 4 表示这个格子为空格。

输入保证这个从格式上来说一定是个合法的 Kakuro 谜题，即每一段连续的空格的左边或者上面的格子包含线索。

接下来 n 行，每行包含若干个正整数，按从左往右的顺序给出初始局面中的每个数字。特别地如果这个格子的种类为 3，那么先给出左下角的线索，再给出右上角的线索。

接下来 n 行，每行包含若干个整数，按从左往右的顺序给出初始局面中的每个数字对应的代价。如果代价为 -1 表示这个格子不能修改，否则代价为非负整数。注意 3 号格子的两个线索有着两个不同的代价。

样例 1 给出了上面的谜题的输入，请在做题前阅读样例 1 确保你理解了输入格式。

Output

一个整数表示最小的代价，如果不可能输出 -1。

Example

kakuro.in	kakuro.out
8 8 0 1 1 0 0 1 1 1 2 4 4 0 3 4 4 4 2 4 4 3 4 4 4 4 2 4 4 4 4 4 1 0 0 2 4 4 3 4 4 1 0 1 3 4 4 4 4 4 2 4 4 4 4 2 4 4 2 4 4 4 0 2 4 4 23 30 27 12 16 16 9 7 17 24 8 7 9 17 8 9 15 29 8 9 5 7 35 6 8 5 9 7 12 7 6 1 7 8 2 6 7 11 10 16 4 6 1 3 2 21 8 9 3 1 5 1 4 6 3 1 2 3 2 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1	0
5 5 0 1 1 1 1 2 4 4 4 4 2 4 4 3 4 2 4 4 4 4 2 4 4 4 4 16 8 6 8 4 4 9 5 4 12 8 4 19 10 4 14 2 3 3 6 1 7 9 4 5 17 5 10 13 11 15 16 4 14 20 20 15 5 16 3 4 3 19 2 4 19 19 13 15 20	822

Constraints

对于 5% 的数据，保证所有的代价都为 -1。

对于 20% 的数据，保证所有空格中的数字代价都为 -1。

对于另外 30% 的数据，保证所有代表线索的数字的代价都为 -1。

对于另外 20% 的数据，保证只有第一行第一列包含线索，剩下的地方全都是空格。

对于 100% 的数据，保证 $3 \le n, m \le 30$，保证初始局面中的每个数字不超过 10^6，保证每个数字的代价不超过 10^6。

Problem B. Cross sum

输入文件: cross.in
输出文件: cross.out
时间限制: 2 second
空间限制: 512 megabytes

游戏规则：

- 空格中填入**正整数**。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之**异或和**，左下角的数字等于其下方邻接之连续方格中数字之**异或和**。
- **所有空格中填入的整数都不能重复**。

Apia 给了 Rimbaud 一个 Kakuro 谜题。心不灵手不巧的 Rimbaud 根本不会做 Kakuro，所以她请求你帮她解决。由于 Rimbaud 很年幼，只会对不超过 $2^{60}-1$ 的数字进行运算。**所以希望谜题的解中每个数字都不超过** $2^{60}-1$。

Input

每组数据包含多组测试数据。第一行包含一个整数 T 表示测试数据组数。

对于每组测试数据，第一行，两个正整数表示 n, m 表示这个游戏的行和列。

接下来 n 行，每行包含 m 个 0 到 4 的数字，第 i 行第 j 列表示第 i 行第 j 列格子的种类。

- 0 表示这个格子既不是空格也不是线索。
- 1 表示这个格子左下角包含线索，右上角没有线索。
- 2 表示这个格子右上角包含线索，左下角没有线索。
- 3 表示这个格子左下角右上角都包含线索。
- 4 表示这个格子为空格。

输入保证这个从格式上来说一定是个合法的 Kakuro 谜题，即每一段连续的空格的左边或者上面的格子包含线索。

接下来 n 行，每行包含若干个**非负整数**，按从左往右的顺序给出谜题中的每个线索。特别地如果这个格子的种类为 3，那么先给出左下角的线索，再给出右上角的线索。

Output

对于每组测试数据，如果有解，那么输出 n 行，每行按从左往右的顺序输出往空格中填入的数，要求空格内填入的数字在 1 到 $2^{60}-1$ 之间。否则输出一个 -1。

Example

cross.in	cross.out
3 3 3 0 1 1 2 4 4 2 4 4 3 7 2 6 3 3 0 1 1 2 4 4 2 4 4 1 1 1 1 2 2 0 1 2 4 0 0	 1 3 2 4 -1 -1

Constraints

对于 10% 的数据，保证 $n, m \le 3$。

对于 30% 的数据，保证 $n, m \le 15$。

对于 50% 的数据，保证 $n, m \le 40$。

对于另外 20% 的数据，保证只有第一行第一列包含线索，剩下的地方全都是空格。

对于 100% 的数据，保证 $3 \le n, m \le 200, T \le 5$，保证初始局面中的每个数字不超过 $2^{60} - 1$。

Problem C. Kreuzsummen

输入文件: kreuz.in
输出文件: kreuz.out
时间限制: 2 second
空间限制: 1024 megabytes

游戏规则：

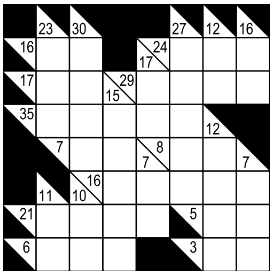
- 方形空格中填入 $\mathbf{1} \sim \mathbf{k}$ 的整数。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之和，左下角的数字等于其下方邻接之连续方格中数字之和。
- 无论是横向还是纵向，连续方格中的数字不能重复。

Apia 给了 Rimbaud 一个 Kakuro 谜题。心不灵手不巧的 Rimbaud 发现自己总是做不出 Kakuro。她怀疑原因是 Apia 给的题目难度太高而不是自己太菜。所以她想评价一下这个题目的难度。

Rimbaud 认为一个 Kakuro 谜题的难度为每个空格的候选数个数的和。我们来定义一下对于每个空格的候选数，这里的候选数为只考虑这个格子在初始谜题下对应的行和列线索和数字不能重复的条件下可以填的数字集合。

如果某连续 4 个空格对应的线索为 10，那么这四个数只可能是 $1+2+3+4$，也就是这几个空格在这个线索的限制下候选数集合为 $\{1,2,3,4\}$。

更一般的，如果某连续 c 个空格对应的线索为 s，那么考虑所有数值 $1 \sim k$ 之间，不重复并且和为 s 的 c 元组，这几个空格在这个线索限制下的候选数集合为出现在这些 c 元组中的所有数字。对于每个空格，它的候选数集合为行限制下的候选数和列限制下的候选数的交。



在这个谜题中，我们考虑第 2 行第 2 列的空格。这个空格的行线索为 16，对应的候选数集合为 $\{7,9\}$。这个空格的列线索为 23，对应的候选数 $\{6,8,9\}$，所以这个空格的候选数集合为 $\{9\}$。第 2 行第 3 列的空格的行线索对应的候选数集合为 $\{7,9\}$，列线索的 I 应的候选数集合为 $\{6,7,8,9\}$，所以这个空格的候选数集合为 $\{7,9\}$。注意虽然我们可以通过先确定第 2 行第 2 列的数字，然后根据行线索推算出第 2 行第 3 列的数字，但是 Rimbaud **只会考虑初始的线索**。

请帮助 Rimbaud 求出这个谜题的难度即所有的空格的候选数个数的和。

Input

第一行，四个正整数表示 n,m,k,T 表示这个游戏的行，列，数值范围和总线索个数。

接下来 T 行，每行五个正整数，$t,x,y_1,y_2,s(t \in \{0,1\}, y_1 \le y_2)$。其中 t 表示这个线索的种类，如果 $t=0$，那么表示这个线索为行线索，第 x 行第 y_1 列到 y_2 列之间的数字和为 s。如果 $t=1$，那么表示这个线索为行线索，第 x 列第 y_1 行到 y_2 行之间的数字和为 s。数据中的行和列都从 1 开始标号。

这个谜题的空格为所有线索对应的空格的并集。输入保证这个从格式上来说一定是个合法的 Kakuro 谜题，即每一段连续的空格的左边或者上面的格子包含线索，并且每个空格出现了恰好两次。

数据中可能出现某些位置的候选数集合为空或者无解的情况。对于这种情况还是只需要按定义求出这个谜题的难度即可。

Output

输出一个整数表示答案。

Example

kreuz.in	kreuz.out
8 8 9 24 0 2 2 3 16 0 2 6 8 24 0 3 2 3 17 0 3 5 8 29 0 4 2 6 35 0 5 3 4 7 0 5 6 7 8 0 6 4 8 16 0 7 2 5 21 0 7 7 8 5 0 8 2 4 6 0 8 7 8 3 1 2 2 4 23 1 2 7 8 11 1 3 2 5 30 1 3 7 8 10 1 4 4 8 15 1 5 3 4 17 1 5 6 7 7 1 6 2 6 27 1 7 2 3 12 1 7 5 8 12 1 8 2 3 16 1 8 6 8 7	127 // 下面为这个样例的解释。 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 2 -1 -1 3 3 2 -1 2 2 -1 2 4 4 2 -1 3 4 1 2 5 -1 -1 -1 -1 1 5 -1 6 5 -1 -1 -1 -1 4 5 5 5 3 -1 8 8 5 6 -1 4 3 -1 2 3 3 -1 -1 2 2

Constraints

对于 10% 的数据，保证 $n, m \leq 3$。

对于 30% 的数据，保证 $n, m \leq 50$。

对于 50% 的数据，保证 $n, m \leq 500$。

对于另外 20% 的数据，保证只有第一行第一列包含线索，剩下的地方全都是空格。

对于 100% 的数据，保证 $3 \leq n, m, T \leq 10^5, 1 \leq k \leq 10^5, s \leq 10^{18}$。