

2023 年全国青少年信息学奥林匹克联赛

CQBZ NOIP 2023

第一试

时间: 2023 年 10 月 26 日 08:00 ~ 12:30

题目名称	下棋	鬼渊	装饰	挖矿
题目类型	传统型	传统型	传统型	传统型
目录	chess	village	decoration	mining
可执行文件名	chess	village	decoration	mining
输入文件名	chess.in	village.in	decoration.in	mining.in
输出文件名	chess.out	village.out	decoration.out	mining.in
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	10	10	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	chess.cpp	village.cpp	decoration.cpp	mining.cpp
-----------	-----------	-------------	----------------	------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项与提醒 (请选手务必仔细阅读)

1. 选手提交的源程序必须存放在已建立好的, 且带有**样例文件**和**下发文件**的文件夹中, 文件夹名称与对应试题英文名一致。
2. 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int, 值必须为 0。
4. **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。**
5. 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**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大, 但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明, 输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格, 请选手使用更完善的读入方式 (例如 `scanf` 函数) 避免出错。

11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

下棋 (chess)

【题目描述】

羽入和小梨花在 $n * m$ 的棋盘上下棋，其中行从 1 到 n 标号，列从 1 到 m 标号。

棋盘上有两种棋子：车和兵。车每次可以往四个方向走任意格，直到在某个空地停下或者碰到一个兵把兵吃掉并停下来。兵每次可以往四个方向走一格，要求格子是空地或者是车，如果是车则将车吃掉。

现在羽入控制车，梨花控制兵，车初始在 (x_1, y_1) ，终点在 (x_2, y_2) ，如果车走到了终点（无论下一步是否存活）则羽入获胜，否则若车被吃了或者羽入在 10^{100000} 步内无法获胜则梨花获胜。双方每次只能移动自己的某一个棋子走一步（也可以选择不移动）。

棋盘初始只有一个车，梨花在开始前可以选择在**任意非车位置**（包括终点）放兵，接着羽入先手操作车，之后两人轮流操作。因为羽入除了会啊啊啊啊以外还很聪明，所以梨花想知道自己需要放的**最少**兵的数量，使得两人都按照最优策略进行时自己可以获胜，如果无解输出 -1 。

【输入格式】

从文件 `chess.in` 中读入数据。

一行六个数 n, m, x_1, y_1, x_2, y_2 ，分别表示棋盘的大小、车的起始位置、终点位置。

【输出格式】

输出到文件 `chess.out` 中。

一行一个数表示答案，无解输出 -1 。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
```

【样例 1 解释】

在 $(1, 2), (2, 2)$ 处放两个兵，显然车无法到达终点。

【数据范围】

对于 100% 的数据保证起点终点不相同, $n, m \leq 300, 1 \leq x_1, x_2 \leq n, 1 \leq y_1, y_2 \leq m$ 。

测试点编号	$n, m \leq$
1 ~ 3	3
4 ~ 6	50
7 ~ 10	300

鬼渊 (village)

【题目描述】

雏见泽是一个和平宁静的小村庄，但是在许多年前这里并不如现在那么和平。从沼泽中涌出的袭击人的鬼，以及流着“鬼之血”的村人们，使得其有了“鬼渊村”的称号。

尽管多年过去，鬼渊村也为了人们口中的传说，但种种线索表明鬼渊村的传说并非空穴来风，并且现在的雏见泽很可能保留着鬼渊村的部分习俗。因此，梨花打算出发寻找鬼渊村旧址，以此来追寻失落的真相。

雏见泽可以看作一个 $n * m$ 的网格图，其中有若干位置可能是曾经鬼渊村的住户，在网格图上用黑色来标明，其余的格子用白色表示。由于雏见泽是鬼渊村扩建形成的，所以鬼渊村是雏见泽中一块矩形区域，并且显然要有至少一户人家。

经过梨花的进一步调查，在当时由于鬼渊的传说，使得在村外的人对其极为恐惧和排斥。因此鬼渊村的村民为了自保全村上下团结一致，连住房都紧密相连，所以只有一个由房屋组成的四连通块。同时为了充分利用空间，不存在若干房屋把一个空地围起来的情况，即连通块内部不存在空地。

现在梨花搞到了雏见泽的地图，在出发之前她想知道有多少个地方可能是鬼渊村旧址。

简要题意：给出一个网格图，每个格子有黑白二色，求有多少个子矩形满足将其挖出来后**恰好**有一个黑色四连通块且不存在由白色格子组成的空腔。

空腔：某个白色格子在空腔内当且仅当其不能通过上下左右四方向走到边界。

【输入格式】

从文件 *village.in* 中读入数据。

第一行两个数 n, m ，表示网格图的大小。

接下来 n 行每行一个长为 m 的 01 串，表示网格图的颜色情况（0 为白 1 为黑）

【输出格式】

输出到文件 *village.out* 中。

一行一个数表示答案。

【样例 1 输入】

```
1 4 4
2 1111
3 1101
4 1001
5 1111
```

【样例 1 输出】

1 83

【样例 2】

见下发文件。

【数据范围】对于全部数据, $0 < n, m \leq 300$ 。

测试点编号	$n \leq$	$m \leq$	特殊性质
1, 2	5	5	无
3, 4	20	20	
5, 6	300	300	保证没有两个黑色格子相邻
7, 8	300	300	保证任意子矩形不存在空腔
9, 10	300	300	无

装饰 (decoration)

【题目描述】

小 F 买了一颗圣诞树，这颗圣诞树是一颗有 n 个节点的树。

为了装饰圣诞树，小 F 又买了 k 种装饰品，准备在 $n-1$ 条边的每条边选择某一种装饰品挂上 (每种装饰品可以重复使用)。

一颗完美的圣诞树肯定要从任何角度看都丰富多彩，小 F 定义了 m 条观察路线，每条路线小 F 会从 a_i 点出发，沿着简单路径观察到 b_i ，如果观察路径上的边至少使用了两种不同的装饰品，她就会认为这个角度观察的圣诞树装饰是丰富多彩的。

请告诉小 F 一共可能有多少种装饰方法满足从 m 个观察路线看都是丰富多彩的，答案可能很大，请对 $10^9 + 7$ 取模。

【输入格式】

从文件 *decoration.in* 中读入数据。

第一行两个正整数 n, m, k ，分别表示树的节点数 n ，装饰品种数 k ，观察路径数 m 。

接下来 $n-1$ 行每行给出两个整数 a_i, b_i ，表示树上有一条边 a_i, b_i 。

之后的 m 行，每行两个数 a_i, b_i ，表示有一条观察路径从 a_i 到 b_i 。

【输出格式】

输出到文件 *decoration.out* 中。

输出一个整数，表示合法的装饰方法数，答案可能很大，请对 $10^9 + 7$ 取模。

【样例 1 输入】

```
1 3 1 2
2 1 2
3 2 3
4 1 3
```

【样例 1 输出】

```
1 2
```

【样例 1 解释】

因为 $1 \rightarrow 2, 2 \rightarrow 3$ 两条边要不能同色，所以只能一个颜色 1 一个颜色 2，所以只有两种可能性。

【样例 2】

见下发文件。

【数据范围】

对于全部数据, $n \leq 60, m \leq 15, k \leq 10^9$ 。

测试点编号	特殊性质
1	$m = 1$
2, 3	每条树边最多只属于 m 条路径中的一条
4, 5	$1 \leq n \leq 15, k = 2$
6 ~ 10	$n \leq 60, m \leq 15, k \leq 10^9$

挖矿 (mining)

【题目描述】

2023 年 10 月, Terraria 发布了 1.9 版本更新。

更新日志中显示, 这个版本添加了一种新的机关, 占用 $n \times m$ 的矩形区域。将这个区域中第 i 行第 j 列的方块记作 $\langle i, j \rangle$ 。

每个方块有一个在 1 到 nm 之间的权值, 记作 $w_{\langle i, j \rangle}$, 所有方块的权值互不相同。你可以选定两个参数 l, r , 满足 $1 \leq l \leq r \leq nm$ 。在此参数作用下, 所有权值在 $[l, r]$ 外的方块将会虚化, 只留下所有权值在 $[l, r]$ 内的方块。形式化地说, 一个方块 $\langle i, j \rangle$ 会被保留当且仅当 $l \leq w_{\langle i, j \rangle} \leq r$ 。

你发现 1.9 版本仍兼容七十年前已停止更新的 tModLoader v1.23.7。你高兴地载入修改日期为 2023/10/26 8:00 的 VeinMiner.tmod 一键挖矿 Mod, 想要试试它能不能对新的机关起作用。

一键挖矿 Mod 可以一次性采集所有与初始挖掘方块四连通的未虚化的方块。也就是说, 可以利用这个 Mod 采集所有的与初始挖掘方块在同一四连通块内的方块。

但是因为 Terraria 1.9 对方块更新进行了优化, 所以这个 Mod 有一个 bug: 如果所有与初始挖掘方块四连通的方块没有形成一个矩形区域, 则无法完整地把这些方块全部采集下来。

你想知道有多少种选择参数 l, r 的方法, 使得在参数作用下, 能够使用一键挖矿 Mod 在不触发 bug 的情况下一次性采集所有未虚化的机关方块。

【输入格式】

从文件 *mining.in* 中读入数据。

第一行两个正整数 n, m , 意义见题目描述。

接下来 n 行, 第 i 行一共 m 个正整数 $w_{\langle i, 1 \rangle} \sim w_{\langle i, m \rangle}$ 表示第 i 行的方块的权值。保证所有方块的权值互不相同。

【输出格式】

输出到文件 *mining.out* 中。

输出仅一行一个整数表示选择参数的方案数。

【样例 1 输入】

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

【样例 1 输出】

1 13

【样例 1 解释】

选择 $\langle l, r \rangle$ 的 13 种方案如下：

形成 1×1 的矩形区域： $\langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle, \langle 4, 4 \rangle, \langle 5, 5 \rangle, \langle 6, 6 \rangle$ 。

形成 1×2 的矩形区域： $\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 4, 5 \rangle, \langle 5, 6 \rangle$ 。

形成 1×3 的矩形区域： $\langle 1, 3 \rangle, \langle 4, 6 \rangle$ 。

形成 2×3 的矩形区域： $\langle 1, 6 \rangle$ 。

【样例 2 输入】

```
1 4 4
2 4 1 5 6
3 3 13 7 11
4 2 14 8 9
5 16 15 12 10
```

【样例 2 输出】

1 32

【样例 3】

见下发文件，该样例的数据类型与最终测试点 11 ~ 12 一致。

【数据范围】

对于全部数据， $1 \leq w_{\langle i, j \rangle} \leq nm \leq 2 \times 10^5$ 。

保证所有 $w_{\langle i, j \rangle}$ 互不相同。

本题评测采用子任务方式，必须通过子任务下的所有测试点才能获得该子任务的分数。保证一个子任务下的测试点个数为 3 个，共计 60 个测试点。

测试点编号	$n \leq$	$m \leq$	$w_{\langle i, j \rangle} = m(i - 1) + j$
1 ~ 2	1	2000	不保证
3 ~ 6	1	2×10^5	不保证
7 ~ 10	2	2×10^5	不保证
11 ~ 12	100	100	不保证
13 ~ 14	2×10^5	2×10^5	保证
15 ~ 20	2×10^5	2×10^5	不保证