

NOIP 模拟赛

CDQZ

时间：2024 年 11 月 18 日

题目名称	亚瑟	巴里	可爱	道格拉斯
题目类型	传统型	传统型	传统型	传统型
可执行文件名	arthur	barry	cute	douglas
输入文件名	arthur.in	barry.in	cute.in	douglas.in
输出文件名	arthur.out	barry.out	cute.out	douglas.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	3.5 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	20	5	8	20
测试点是否等分	是	否	否	是

提交源程序文件名

对于 C++ 语言	arthur.cpp	barry.cpp	cute.cpp	douglas.cpp
-----------	------------	-----------	----------	-------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 选手提交的源程序请直接放在个人目录下，无需建立子文件夹；
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. 若无特殊说明，每道题的代码大小限制为 100KB。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等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。

14. 统一评测时采用的机器配置为：12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz, 内存 16GB。上述时限以此配置为准。

15. 评测在 Windows 11 下进行，使用 LemonLime 进行评测。如果对此条以及 14 条中的机器配置有疑问，请及时询问。

16. 本次比赛的难度较为简单，请 AK 的选手不要大声喧哗。

亚瑟 (arthur)

【题目描述】

小 L 是一个喜欢研究树的小孩，他的树有 n 个节点，编号为 $1, 2, \dots, n$ 。定义 $f(l, r)$ 为仅保留编号为 $l \sim r$ 的点和两个端点编号都在 $[l, r]$ 里的边的连通块数。小 L 希望求出 $\sum_{l=1}^n \sum_{r=l}^n f(l, r)$ ，小 L 希望你帮助他解决这个问题。

【输入格式】

- 从文件 `arthur.in` 中读入数据。
- 第一行输入一个正整数 n ，表示树的节点数。
- 第二行到第 $n + 1$ 行，每行输入两个数 u, v ，表示 u, v 之间连接一条无向边。

【输出格式】

- 输出到文件 `arthur.out` 中。
- 输出一个数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 7
```

【样例 2 输入】

```
1 10
2 5 3
3 5 7
4 8 9
5 1 9
6 9 10
7 8 4
```

874

9610

1072

【样例 2 输出】

1113

【样例 3】

见选手目录下的 *arthur/arthur3.in* 与 *arthur/arthur3.ans*。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 10^6, 1 \leq u, v \leq n$, 且输入数据构成一棵树。

测试点编号	$n \leq$	特殊性质	分值
1 ~ 4	200	无	20
5 ~ 6	10^6	A	10
7 ~ 8		B	10
9 ~ 14	4000	无	30
15 ~ 20	10^6		30

特殊性质 A : $\forall i > 1, i$ 与 $i - 1$ 有连边。
特殊性质 B : 存在一个点 u , 使得所有的 $v \neq u$ 都满足 u 和 v 有连边。

巴里 (barry)

【题目描述】

小 W 喜欢解密，有一个 $n \times m$ 的网格图，每个位置有一个 $1 \sim p$ 的数字 $a_{i,j}$ ，代表这个位置有一个线索 $a_{i,j}$ ，如果你走到了一个线索为 x 的格子，且已经获得了 $1 \sim x - 1$ 的所有线索，那么你就可以获得线索 x 。

小 W 初始在 $(1,1)$ ，每一步可以走到相邻的四个点中的一个（不能走到网格图外面），初始时还没有任何线索，所以只能去获得线索 1。现在小 W 想知道收集完所有种类的线索最少需要走多少步。保证网格图中出现了 $1 \sim p$ 中的所有数。

【输入格式】

从文件 `barry.in` 中读入数据。

第一行三个正整数 n, m, p ，表示这是一个 $n \times m$ 的网格图， p 表示线索的种类数。

接下来 n 行，每行 m 个正整数，表示 n 行 m 列的网格图。第 $i + 1$ 行的第 j 个正整数表示 $a_{i,j}$ 。

【输出格式】

输出到文件 `barry.out` 中。

输出一行一个数，表示收集完所有的线索最少需要多少步。

【样例 1 输入】

```
1 3 3 9
2 1 3 5
3 8 9 7
4 4 6 2
```

【样例 1 输出】

```
1 22
```

【样例 1 解释】

定义 $(a,b) \rightarrow (c,d)$ 表示从 (a,b) 到 (c,d) 的最短路径。

$(1,1) \rightarrow (3,3) \rightarrow (1,2) \rightarrow (3,1) \rightarrow (1,3) \rightarrow (3,2) \rightarrow (2,3) \rightarrow (2,1) \rightarrow (2,2)$ 为最优路径。

【样例 2 输入】

```
1 3 4 12
2 1 2 3 4
3 8 7 6 5
4 9 10 11 12
```

【样例 2 输出】

```
1 11
```

【样例 3】

见选手目录下的 *barry/barry3.in* 与 *barry/barry3.ans*。
该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *barry/barry4.in* 与 *barry/barry4.ans*。
该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *barry/barry5.in* 与 *barry/barry5.ans*。
该样例满足子任务 5 的限制。

【数据范围】

对于所有数据，保证 $1 \leq n, m \leq 700, 1 \leq a_{i,j} \leq p \leq nm$ ，且网格图中出现了 $1 \sim p$ 中的所有数。设 w_i 表示线索 i 的数量。

子任务编号	特殊性质	分值
1	$w_i = 1$	10
2	$nm \leq 5000$	18
3	$\forall i \in [1, p), w_i \times w_{i+1} \leq nm$	22
4	$nm \leq 2 \times 10^5$	33
5	无	17

可爱 (cute)

【题目描述】

小 H 是一个 cute 的人，同样，他特别喜欢 cute 的东西。现在他有 k 棵树，编号为 $1 \sim k$ 。每棵树的大小都为 n ，点的编号为 $0 \sim n - 1$ 。

定义函数 $\text{cute}(t, x, y)$ ，表示第 t 棵树上从点 x 到点 y 的简单路径上的点的编号的 mex 值。

现在小 H 想知道 k 棵树到底有多 cute，所以想请你求出下面式子的答案，用来表示 k 棵树的总 cute 值：

$$\sum_{x=0}^{n-1} \sum_{y=0}^{n-1} \max_{t=1}^k \text{cute}(t, x, y)$$

定义 $\text{mex}(S)$ 表示自然数集合 S 中最小的没有出现过的自然数。

【输入格式】

从文件 `cute.in` 中读入数据。

第一行两个正整数 n, k 表示共有 k 棵树，每棵树大小为 n 。

接下来 $k \times (n - 1)$ 行，每 $n - 1$ 行表示一棵树。其中每行两个非负整数 u, v 表示 u 与 v 之间有一条无向边连接。

【输出格式】

输出到文件 `cute.out` 中。

输出一行一个数，表示这 k 棵树的总 cute 值。

【样例 1 输入】

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

【样例 1 输出】

```
1 11
```

【样例 2 输入】

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

【样例 2 输出】

```
1 19
```

【样例 3】

见选手目录下的 *cute/cute3.in* 与 *cute/cute3.ans*。
该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *cute/cute4.in* 与 *cute/cute4.ans*。
该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *cute/cute5.in* 与 *cute/cute5.ans*。
该样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *cute/cute6.in* 与 *cute/cute6.ans*。
该样例满足子任务 5 的限制。

【样例 7】

见选手目录下的 *cute/cute7.in* 与 *cute/cute7.ans*。
该样例满足子任务 6 的限制。

【样例 8】

见选手目录下的 *cute/cute8.in* 与 *cute/cute8.ans*。
该样例满足子任务 7 的限制。

【样例 9】

见选手目录下的 *cute/cute9.in* 与 *cute/cute9.ans*。
该样例满足子任务 8 的限制。

【数据范围】

对于所有数据，保证 $3 \leq n \leq 3 \times 10^5, 1 \leq k \leq 3, 0 \leq u, v < n$ ，且输入数据构成 k 棵树。

子任务编号	$n \leq$	k	特殊性质	分值
1	3	≤ 3	无	3
2	30			7
3	3000			10
4	3×10^5	$= 1$		15
5		$= 2$	有	15
6			无	15
7		$= 3$	有	15
8			无	20

特殊性质：保证数据随机生成，生成方式为：随机生成一个 $0 \sim n - 1$ 的排列，除了第一个点外，每个点随机向前面的一个点连边。

道格拉斯 (douglas)

【题目描述】

在一张 $n \times n$ 的网格图中，有 m 个矩形障碍，障碍之间可能有交。你要从左下角 $(1, 1)$ 走到右上角 (n, n) ，每次只能向上或者向右走一格，不能经过障碍，求有多少种本质不同的走法。

两个走法本质不同，当且仅当存在一个障碍，一个走法是从上方绕过它，一个走法是从下方绕过它。

【输入格式】

从文件 *douglas.in* 中读入数据。

第一行两个正整数 n, m ，分别表示棋盘大小和障碍数。

接下来 m 行，每行四个正整数 x_1, y_1, x_2, y_2 ，表示一个矩形障碍的左下角坐标和右上角坐标。

【输出格式】

输出到文件 *douglas.out* 中。

输出一行一个正整数，表示本质不同的走法数。

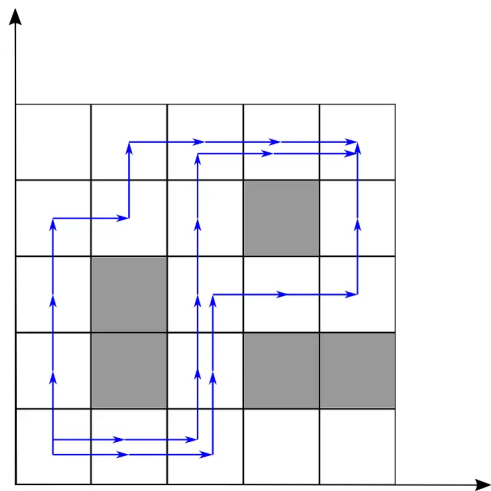
【样例 1 输入】

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

【样例 1 输出】

```
1 3
```

【样例 1 解释】



共有如上三种本质不同的走法。

【样例 2 输入】

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

【样例 2 输出】

```
1 3
```

【样例 3】

见选手目录下的 *douglas/douglas3.in* 与 *douglas/douglas3.ans*。
该样例满足测试点编号 1,2 的限制。

【样例 4】

见选手目录下的 *douglas/douglas4.in* 与 *douglas/douglas4.ans*。
该样例满足测试点编号 5~9 的限制。

【样例 5】
见选手目录下的 *douglas/douglas5.in* 与 *douglas/douglas5.ans*。
该样例满足测试点编号 12 ~ 14 的限制。

【样例 6】
见选手目录下的 *douglas/douglas6.in* 与 *douglas/douglas6.ans*。
该样例满足测试点编号 17 ~ 20 的限制。

【数据范围】
对于所有数据，保证 $3 \leq n \leq 10^6, 1 \leq m \leq 10^6, 1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq n$ 。保证 $(1, 1)$ 和 (n, n) 处没有障碍物。

测试点编号	$n \leq$	$m \leq$	特殊性质
1, 2	10	10	无
3, 4	30		
5 ~ 9	1000	400	
10, 11	10^6		
12 ~ 14		1000	有
15, 16		10^5	
17 ~ 20		10^6	无

特殊性质：保证任意两个矩形不交。