

0814 模拟赛

时间：2026 年 8 月 14 日 8:30~12:00

题目名称	道路维护	巅峰对决	密令备份	三色
题目类型	传统型	传统型	传统型	传统型
目录	road	clim	cipher	tricolor
输入文件名	road.in	clim.in	cipher.in	tricolor.in
输出文件名	road.out	clim.out	cipher.out	tricolor.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	2.0 秒
内存限制	256 MiB	256 MiB	256 MiB	256 MiB
测试点数目	10	20	20	20
子任务数目	1	1	1	4
测试点是否等分	是	是	是	否

提交源程序文件名

对于 C++ 语言	road.cpp	clim.cpp	cipher.cpp	tricolor.cpp
-----------	----------	----------	------------	--------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static -Wl,--stack,536870912
-----------	--

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. `main` 函数的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 若无特殊说明，结果比较方式为全文比较（过滤行末空格及文末回车）。
4. 选手提交的源程序文件不得大于 **100KB**。
5. 提交路径为 选手姓名/题目目录/程序源文件，例如 张三/problem1/problem1.cpp。
6. 程序可使用的栈空间限制与题目内存限制一致。
7. 评测环境为 **Windows**，CPU 为 **AMD Ryzen R9 8945HS**，内存 **32GB**。
8. 禁止在源代码中修改编译器参数（如使用 `#pragma` 指令），禁止使用与系统结构相关的指令和其他可能造成不公平的方法。

道路维护 (road)

【题目描述】

小兔国有 n 座城市，编号为 1 到 n 。城市之间有 $n - 1$ 条双向道路，任意两座城市之间都可以通过这些道路互相到达。

现在，小兔国需要对其中恰好 k 条道路进行维护。在维护期间，这些道路将被封闭，车辆无法通行。

对于每座城市 u ，定义它的便利度为从城市 u 出发，仅经过未被封闭的道路所能到达的城市数量。特别地，城市 u 自身也计入其中。

整个国家的总便利度等于所有城市便利度之和。小兔国可以自行选择需要维护的 k 条道路，请求出总便利度的最大值。

【输入格式】

从文件 `road.in` 中读入数据。

第一行包含两个整数 n, k ，分别表示城市的数量和需要封闭的道路数量。

接下来 $n - 1$ 行，每行包含两个整数 u, v ，表示城市 u 和城市 v 之间有一条双向道路。

保证给出的道路构成一棵树。

【输出格式】

输出到文件 `road.out` 中。

输出一个整数，表示总便利度的最大值。

【样例 1 输入】

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

【样例 1 输出】

```
1 11
```

【样例 1 解释】

例如封闭道路 $(1, 2)$ 和 $(4, 5)$ 后，三个连通块的大小分别为 $1, 3, 1$ ，总便利度为 $1^2 + 3^2 + 1^2 = 11$ 。

【样例 2 输入】

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

【样例 2 输出】

```
1 19
```

【样例 3】

见选手目录下的 `road/road3.in` 与 `road/road3.ans`。

【数据范围】

对于所有测试数据，保证 $1 \leq n \leq 2 \times 10^5$ ， $0 \leq k < n$ ， $1 \leq u, v \leq n$ 。
本题共 10 个测试点，每个测试点 10 分。

测试点编号	$n \leq$	$k \leq$	特殊性质
1, 2	10	$n - 1$	无
3	2×10^5	0	无
4	2×10^5	1	A
5	2×10^5	$n - 1$	B
6, 7	2×10^5	$n - 1$	C
8 ~ 10	2×10^5	$n - 1$	无

- 特殊性质 A： $k = 1$ 。
- 特殊性质 B： $k = n - 1$ 。
- 特殊性质 C：每条道路连接的城市 u, v 均满足 $u = v + 1$ 。

巅峰对决 (clim)

【题目描述】

两名登山者——先手玩家和后手玩家——在一张山地地图上博弈。

地图是一个 n 行 m 列的网格。第 i 行第 j 列的格子有一个整数高度 $h_{i,j}$ 。地图上所有高度互不相同。

一个格子上放有一面旗帜。两名玩家交替移动旗帜，先手玩家先手。一名玩家的回合中：

1. 该玩家必须将旗帜从当前格子移动到一个四相邻（共享一条边）且高度严格更高的格子；
2. 如果当前格子不存在这样的格子，则轮到移动的玩家无法移动并输掉游戏，另一方获胜。

由于每次移动都到一个严格更高的格子，旗帜永远不会回到它已经离开的格子，因此游戏总会在有限步之内结束。双方都采取最优策略。

给定 q 个独立的询问。每个询问指定旗帜的起始格子；地图本身在询问之间不会改变。对于每个询问，判断在双方最优博弈下谁会获胜。

【输入格式】

从文件 *clim.in* 中读入数据。

每个测试文件包含多组测试数据。第一行包含测试数据的组数 T 。

每组测试数据的第一行包含两个整数 n, m ，分别表示地图的行数和列数。

接下来 n 行描述高度。其中第 i 行包含 m 个整数 $h_{i,1}, h_{i,2}, \dots, h_{i,m}$ 。保证同一组测试数据内的所有高度互不相同。

接下来一行包含一个整数 q ，表示询问数。

接下来 q 行，每行包含两个整数 r, c ，表示旗帜的起始格子为第 r 行第 c 列。

【输出格式】

输出到文件 *clim.out* 中。

对于每个询问输出一行：如果先手玩家在最优博弈下获胜，则输出 **First**；否则输出 **Second**。

【样例 1 输入】

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

【样例 1 输出】

```
1 First
2 Second
3 Second
4 Second
5 First
```

【样例 1 解释】

格子 (2,1) 的高度为 6，没有更高的相邻格子，因此先手玩家立即输掉。

从格子 (2,3) 出发时，博弈过程被强制为 $4 \rightarrow 5 \rightarrow 6$ ，此时再次轮到先手玩家且无法移动，所以答案也是 Second。

【样例 2】

见选手目录下的 *clim/clim2.in* 与 *clim/clim2.ans*。

【样例 3】

见选手目录下的 *clim/clim3.in* 与 *clim/clim3.ans*。

【数据范围】

对于所有测试数据，保证： $1 \leq T \leq 500$ ； $1 \leq n, m \leq 10^5$ ，单组测试数据中 $n \times m \leq 10^5$ ；同一测试文件中所有测试数据的格子总数不超过 10^5 ； $1 \leq h_{i,j} \leq 10^9$ ，同一组测试数据内所有高度互不相同； $1 \leq q \leq 10^5$ ，同一测试文件中询问总数不超过 10^5 ； $1 \leq r \leq n$ ， $1 \leq c \leq m$ 。

测试点编号	$n \times m \leq$	$q \leq$	特殊性质
1 ~ 3	10^5	10^5	A
4 ~ 6	20	20	无
7 ~ 10	1000	1000	无
11 ~ 14	10^5	10^5	B
15 ~ 20	10^5	10^5	无

特殊性质 A： $n = 1$ 。
特殊性质 B： 每个格子至多有一个四相邻且高度严格更高的格子。

密令备份 (cipher)

【题目描述】

特工兔和兔老大之间的电报需要加密，密令 S 由 n 个大写英文字母组成。兔老大希望设计一个长度恰好为 $2n$ 的新密令 P 作为备份，并要求：

1. P 是回文串，即从左向右和从右向左读完全相同；
2. S 是 P 的子序列。

如果可以从 P 中删除若干个字符（也可以一个都不删除），使剩余字符按照原来的相对顺序组成 S ，则称 S 是 P 的子序列。被删除的字符不要求连续。

请求出满足条件的不同密令 P 的数量。

【输入格式】

从文件 *cipher.in* 中读入数据。

第一行包含一个整数 n ，表示密令 S 的长度。

第二行包含一个长度为 n 的字符串 S ，其中每个字符均为大写英文字母。当 $n = 0$ 时，第二行为空行。

【输出格式】

输出到文件 *cipher.out* 中。

输出一个整数，表示满足条件的不同回文密令数量对 $10^9 + 7$ 取模后的结果。

【样例 1 输入】

1 2

2 AA

【样例 1 输出】

1 51

【样例 2 输入】

1 2

2 AB

【样例 2 输出】

1 2

【样例 3】

见选手目录下的 `cipher/cipher3.in` 与 `cipher/cipher3.ans`。

【样例 4】

见选手目录下的 `cipher/cipher4.in` 与 `cipher/cipher4.ans`。

【数据范围】

对于所有测试数据，保证 $0 \leq n \leq 200$ ， $|S| = n$ ，且 S 仅由大写英文字母组成。
令 c 表示 S 中不同字符的种类数。
本题共 20 个测试点，每个测试点 5 分。

测试点编号	$n \leq$	$c \leq$	特殊性质
1	0	0	无
2	1	1	无
3 ~ 5	4	26	无
6	26	26	A
7	200	1	无
8, 14, 15, 20	200	2	无
9, 11 ~ 13, 19	200	5	无
10	200	26	无
16, 17	200	4	无
18	200	26	B

特殊性质 A： S 中任意两个字符均不相同。
特殊性质 B： S 是回文串。

三色 (tricolor)

【题目描述】

给定一个 $n \times m$ 的网格。你需要在每个格子中填入 $0, 1, 2$ 中的一个整数。
如果任意两个共享一条边的格子中填入的整数都不同，则称这种填法是好的。
请你求出好的填法数量。由于答案可能很大，请输出答案对 998244353 取模后的结果。

【输入格式】

从文件 *tricolor.in* 中读入数据。
一行包含两个整数 n, m 。

【输出格式】

输出到文件 *tricolor.out* 中。
输出一个整数，表示好的填法数量对 998244353 取模后的结果。

【样例 1 输入】

```
1 2 2
```

【样例 1 输出】

```
1 18
```

【样例 1 解释】

第一列共有 6 种合法填法。对于第一列的任意一种合法填法，第二列恰好有 3 种合法填法，因此答案为 $6 \times 3 = 18$ 。

【样例 2】

见选手目录下的 *tricolor/tricolor2.in* 与 *tricolor/tricolor2.ans*。

【样例 3】

见选手目录下的 *tricolor/tricolor3.in* 与 *tricolor/tricolor3.ans*。

【数据范围】

对于所有测试数据，保证 $1 \leq n < 10$, $1 \leq m < 998244353$ 。
本题采用捆绑测试，选手必须通过子任务下的所有测试点才可取得对应测试点的分数。

子任务编号	分值	$n \leq$	$m \leq$	特殊性质
1	10	3	5	无
2	20	9	50	无
3	40	6	998244352	无
4	30	9	998244352	无