



## 提高级

题目名称	黑白棋	魔法大陆	树上修路	破损的棋盘
题目类型	传统型	传统型	传统型	传统型
目录	chess	magic	treeroad	chessboard
可执行文件名	chess	magic	treeroad	chessboard
输入文件名	chess.in	magic.in	treeroad.in	chessboard.in
输出文件名	chess.out	magic.out	treeroad.out	chessboard.out
每个测试点时限	1000ms	1000ms	1000ms	1000ms
内存限制	512mb	512mb	512mb	512mb
子任务数目	10	10	10	10
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	magic.cpp	treeroad.cpp	chessboard.cpp
---------	-----------	-----------	--------------	----------------

编译选项

对于C++语言	-O2 -lm
---------	---------

### 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. C/C++中函数main()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，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈空间内存限制与题目的内存限制一致。
5. 只提供Linux格式附加样例文件。

# 黑白棋 (chess)

## 题目描述

小 Z 迷上了棋类游戏，聪明的他很快就摸清了黑白棋、五子棋和围棋的规则，但今天 Prof. Wang 给了他一项挑战：需将杂乱的棋子按黑白相间的特定序列排列。小 Z 决定通过交换棋子位置的方式将它们排列整齐，对于位置分别为  $i, j$  的两颗棋子，需要花费  $|i - j|^k$  的代价完成交换。杂乱无章的棋子让小 Z 头疼不已，你能帮帮他，找到让代价最小的交换方式吗？

## 输入格式

第一行包含一个整数  $T$  表示测试数据的种类数

接下来  $2 \cdot T$  行，对于每组测试数据，第一行为一个整数  $k$ ，表示代价函数的类型，第二行一个由 **B** 和 **W** 组成的长度为  $l$  的字符串，分别表示黑色和白色。

## 输出格式

$T$  行，每行输出一个整数，表示最小的总代价。如果不存在一种方案使得棋子黑白相间排列，输出  $-1$ 。

## 样例

1

2

BWBWW

2

### 样例解释 #1

交换 1 位置的黑点和 2 位置的白棋，以及 3 位置的黑棋和 4 位置的白棋，序列变为 WBWBW，代价为 2。可以证明这是代价最小的一种方案。

## 约定

每组测试数据保证  $1 \leq T \leq 5$

对于 20% 的数据， $k = 0, 1 \leq l \leq 10$

对于 40% 的数据， $0 \leq k \leq 1, 1 \leq l \leq 100$

对于 70% 的数据， $0 \leq k \leq 3, 1 \leq l \leq 1000$

对于 100% 的数据， $0 \leq k \leq 10^5, 1 \leq l \leq 10^5$

# 魔法大陆 (magic)

## 背景故事

在一个神秘的魔法大陆上，存在着许多魔法能量点（节点）。这些节点之间的魔法能量可以通过魔法阵连接起来，以增强整个大陆的魔法力量。每个魔法阵可以覆盖一段连续的节点，但每个魔法阵的启动需要花费一定的魔力。

魔法师协会希望至少激活  $k$  个节点，但由于魔法阵之间的能量干扰，每个节点最多只能被  $p_i$  个魔法阵覆盖。现在需要你帮助魔法师协会设计一个方案，使得在满足这些条件的前提下，启动魔法阵的总花费最少。

## 问题描述

在一个长度为  $n$  的区间  $[1, n]$  中，有  $m$  个子区间。每个子区间  $[l_i, r_i]$  代表可以由一个魔法阵覆盖的范围，其覆盖的代价为  $c_i$ 。现在要求覆盖至少  $k$  个节点，并且需要满足第  $i$  个节点最多被覆盖  $p_i$  次。求覆盖这些节点的最小代价。

## 输入格式

- 第一行包含三个整数  $n$ 、 $m$  和  $k$ 。
- 接下来  $m$  行，每行包含三个整数  $l_i$ 、 $r_i$  和  $c_i$ ，表示第  $i$  个子区间  $[l_i, r_i]$  的左右端点和覆盖代价。
- 最后一行包含  $n$  个整数  $p_i$ ，表示第  $i$  个节点最多可以被覆盖  $p_i$  次。

## 输出格式

输出覆盖至少  $k$  个节点且满足第  $i$  个节点最多被覆盖  $p_i$  次的最小代价。如果无法满足条件，输出  $-1$ 。

## 样例

```
5 3 4
1 3 10
2 4 20
3 5 30
3 3 3 3 3
```

30

```
5 3 4
1 3 10
2 4 20
3 5 30
1 1 1 1 1
```

-1

# 提示

---

1.  $1 \leq n \leq 300$

2.  $1 \leq m \leq 10^5$

3.  $1 \leq k \leq n$

4.  $1 \leq l_i \leq r_i \leq n$

5.  $1 \leq c_i \leq 10^9$

6.  $1 \leq p_i \leq m$

7. 对于30%的数据, 所有 $p_i = 1$

# 树上修路 (treeroad)

## 背景：

在古树村，生长着一棵庞大的古树。这棵古树并非普通的树，它的树干和树枝之间形成了无数的节点，节点之间通过树枝相连，构成了一棵巨大的树结构。村里的长老决定，要在这棵古树上修建一套道路系统，方便村民们在树上移动和运输。

长老们制定了两项计划来管理这个修路项目：

1. **修路计划**：当村民们需要在树上的某两个节点之间移动时，他们会提出修路请求，沿着这两个节点之间的路径修建道路。这条道路会沿着连接这两个节点的所有树枝修建。
2. **连通性查询**：为了确保道路系统的有效性，长老们需要时刻了解某两个节点之间的连通情况。他们会查询在这两个节点之间的路径上有多少个连通块。连通块可以是由修建的道路直接连通的一组节点，或者是没有修路而单独存在的节点。

现在，他们需要你的帮助来实现这个修路项目，并且能够准确地回答每一个连通性的查询。

## 题目描述

给出一棵  $n$  个节点的树（编号  $1 \sim n$ ），编号为  $1 \sim n$ 。现有如下两种操作：

1. 对于  $(u, v)$  路径，沿着树的每条边修一条路。
2. 询问  $(u, v)$  路径上有多少个连通块。

注意：单独的一个节点也算一个连通块。

## 输入格式

- 第一行两个整数， $n, m$ （ $m$  次操作）。
- 接下来  $n - 1$  行，分别表示节点 2 到  $n$  的父节点编号。
- 接下来  $m$  行每行三个整数  $op, u, v$ 。 $op = 1$  为操作 1， $op = 2$  为操作 2。

## 输出格式

对每个操作 2 输出一行表示查询的答案。

## 样例

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

```
2
```

## 说明/提示

---

- 对于 30 的数据,  $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^5$ , 数据纯随机。
- 对于 60 的数据,  $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^5$ 。
- 对于 100 的数据,  $1 \leq n \leq 5 \times 10^5$ ,  $1 \leq m \leq 5 \times 10^5$ 。

# 破损的棋盘 (chessboard)

## 题目描述

小 Z 刚刚学习了 N 皇后问题。但相较于国际象棋，他更喜欢中国象棋。

N 皇后问题需要在棋盘上摆放 N 个皇后，使其互相不能攻击，计算符合条件方案数。小 Z 发现了中国象棋中没有皇后，于是决定用来车来替代。因此，他现在想在棋盘上摆放 N 个车，且两两不能互相攻击到。为了弥补没有皇后的斜线控制的缺憾，小 Z 希望在主副对角线上都至少存在一个车。

由于棋盘保存不慎，小 Z 的棋盘上少数位置被虫蛀漏洞了，简单说，有  $m$  个位置上不能再摆放棋子。小 Z 想知道在他的这副棋盘上，有多少种摆放 N 个车的方案。由于结果可能很大，请你输出对 998244353 取模后的结果。

## 输入格式

第一行包含一个整数  $T$  表示测试数据的组数。

接下来  $T$  组测试数据，对于每组测试数据，第一行有两个  $N, m$ ，分别表示棋盘的大小和被虫蛀的格子数量，接下来  $m$  行每行两个整数  $(x_i, y_i)$  表示被漏洞的行列坐标。

## 输出格式

$T$  行，每行输出一个整数，表示方案数对 998244353 取模后的结果。

## 样例

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

```
10
27
99
```

## 约定

每组测试数据保证  $1 \leq T \leq 5, 0 \leq m \leq n^2, 1 \leq x_i, y_i \leq N$

对于 20% 的数据， $1 \leq N \leq 10, 0 \leq m \leq 8$

对于另外 20% 的数据， $1 \leq N \leq 30, m = 0$

对于 100% 的数据， $1 \leq N \leq 100, 0 \leq m \leq 10$

