

# NOIP模拟

题目名称	印章	Deadlock	二叉树	麻将机
题目类型	传统型	传统型	传统型	传统型
目录	seal	deadlock	tree	machine
可执行文件名	seal	deadlock	tree	machine
输入文件名	seal.in	deadlock.in	tree.in	machine.in
输出文件名	seal.out	deadlock.out	tree.out	machine.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	3	7	4	10
测试点是否等分	否	否	否	是

提交源程序文件名

对于 C++ 语言	seal.cpp	deadlock.cpp	tree.cpp	machine.cpp
-----------	----------	--------------	----------	-------------

编译选项

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

## 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。

## 印章 (seal)

### 【题目描述】

一张  $n \times m$  的方格纸，有些格子需要印成黑色，剩下的格子需要保留白色。

你有一个  $a \times b$  的印章，有些格子是凸起（会沾上墨水）的。你需要判断能否用这个印章印出纸上的图案。印的过程中需要满足以下要求：

- 印章不可以旋转。
- 不能把墨水印到纸外面。
- 纸上的同一个格子不可以印多次。

### 【输入格式】

从文件 *seal.in* 中读入数据。

第一行包含 4 个整数  $n, m, a, b$ 。

接下来  $n$  行，每行  $m$  个字符，描述纸上的图案。’.’ 表示留白，’x’ 表示需要染黑。

接下来  $a$  行，每行  $b$  个字符，描述印章。’.’ 表示不沾墨水，’x’ 表示沾墨水。

### 【输出格式】

输出到文件 *seal.out* 中。

输出 TAK（是）或 NIE（否），表示能否用这个印章印出纸上的图案。

### 【样例 1 输入】

```
1 3 4 4 2
2 xx..
3 .xx.
4 xx..
5 x.
6 .x
7 x.
8 ..
```

### 【样例 1 输出】

```
1 TAK
```

### 【样例 2 输入】

```
1 2 2 2 2
2 xx
3 xx
4 .x
5 x.
```

**【样例 2 输出】**

```
1 NIE
```

**【样例 3】**

见附加文件中的 seal/ex\_seal3.in 与 seal/ex\_seal3.ans。

**【数据范围】**

本题采用捆绑测试。

对于全部数据， $1 \leq n, m, a, b \leq 1000$ 。

子任务一（40 分）： $1 \leq n, m, a, b \leq 100$ 。

子任务二（30 分）： $1 \leq n, m, a, b \leq 500$ 。

子任务三（30 分）： $1 \leq n, m, a, b \leq 1000$ 。

# Deadlock (deadlock)

## 【题目描述】

Valorant 总部内, Deadlock 和 Killjoy 正在研究王国集团建立的连接地球  $\omega$  与地球  $\alpha$  的传送门网络, 但杂乱无章的地图似乎对揣测王国集团的下一步意图毫无用处。

Killjoy 通过演算发现, 整个网络总共有  $n$  个节点, 节点 1 表示的是地球  $\omega$  的坐标, 而节点  $n$  是地球  $\alpha$  的坐标, 其他的节点是无关紧要的王国集团传送中枢或防御工事。

每个节点都有自己的传送门, 每两个不同的传送门间都有且只有一条传送链接, 传送链接是双向开放的。但 Deadlock 指出, 只有一些传送链接一开始是打开的。Killjoy 和 Deadlock 都很不解, 于是 Killjoy 派 Deadlock 去找线人 Cypher 先生聊聊。在 Deadlock 献出了整个组织库存的所有草莓蛋糕后, Cypher 同意和 Deadlock 玩个游戏。Cypher 打开了给 Killjoy 的地图的副本。然后他制定了游戏规则。

- 每次游戏开始时, Cypher 会拿出一张副本。
- 先由 Cypher 选择一个关闭的传送链接打开。
- 再由 Deadlock 选择一个关闭的传送链接打开。
- 注意每人操作完后必须保证地球  $\omega$  和  $\alpha$  不能互相到达。若轮到某一方操作时无法进行操作, 则这一方输。

Deadlock 并不理解 Cypher 先生在暗示什么, 但她选择先陪 Cypher 进行游戏。现在给出总计  $t$  次游戏的初始副本, 对于每次游戏, 已知副本内有  $n$  个节点, 有  $m$  个传送链接已开启, 其中第  $i$  个开启的传送链接位于  $u_i, v_i$  之间。要求你判断在两人无敌聪明的情况下谁会赢。

## 【输入格式】

从文件 *deadlock.in* 中读入数据。

第一行包含一个正整数  $t$ , 表示总共进行了多少次游戏。

对于每轮游戏:

- 第一行, 两个正整数  $n, m$ , 表示此次游戏的地图副本有  $n$  个节点, 有  $m$  个已开启的传送链接。
- 接下来  $m$  行, 每行两个正整数  $u_i, v_i$ , 表示第  $i$  个开启的传送链接位于  $u_i, v_i$  之间。

## 【输出格式】

输出到文件 *deadlock.out* 中。

输出共  $t$  行, 每行包含一个字符串。若 Cypher 会赢, 请输出 Cypher; 同理, 若 Deadlock 会赢, 请输出 Deadlock。

**【样例 1 输入】**

```
1 3
2 3 0
3 6 2
4 1 2
5 2 3
6 15 10
7 12 14
8 8 3
9 10 1
10 14 6
11 12 6
12 1 9
13 13 1
14 2 5
15 3 9
16 7 2
```

**【样例 1 输出】**

```
1 Cypher
2 Deadlock
3 Cypher
```

**【样例 2】**

见附加文件中的 deadlock/ex\_deadlock2.in 与 deadlock/ex\_deadlock2.ans。

**【样例 3】**

见附加文件中的 deadlock/ex\_deadlock3.in 与 deadlock/ex\_deadlock3.ans。

**【数据范围】**

本题采用捆绑测试。

对于全部数据,  $1 \leq t \leq 10^5, 2 \leq n \leq 10^5, 0 \leq m \leq \min(\frac{n(n-1)}{2}, 10^5), 1 \leq u_i, v_i \leq n, \sum n, \sum m \leq 2 \times 10^5$ 。

子任务一 (5 分):  $n = 2, t = 1$ 。

子任务二 (5 分):  $n = 3, t = 1$ 。

子任务三 (5 分):  $n = 5, t = 2$ 。

子任务四 (5 分):  $n = 10, t = 1$ 。

子任务五 (5 分):  $n = 20, t = 1$ 。

子任务六 (25 分):  $n = 10^2, 0 \leq m \leq 10^2$ 。

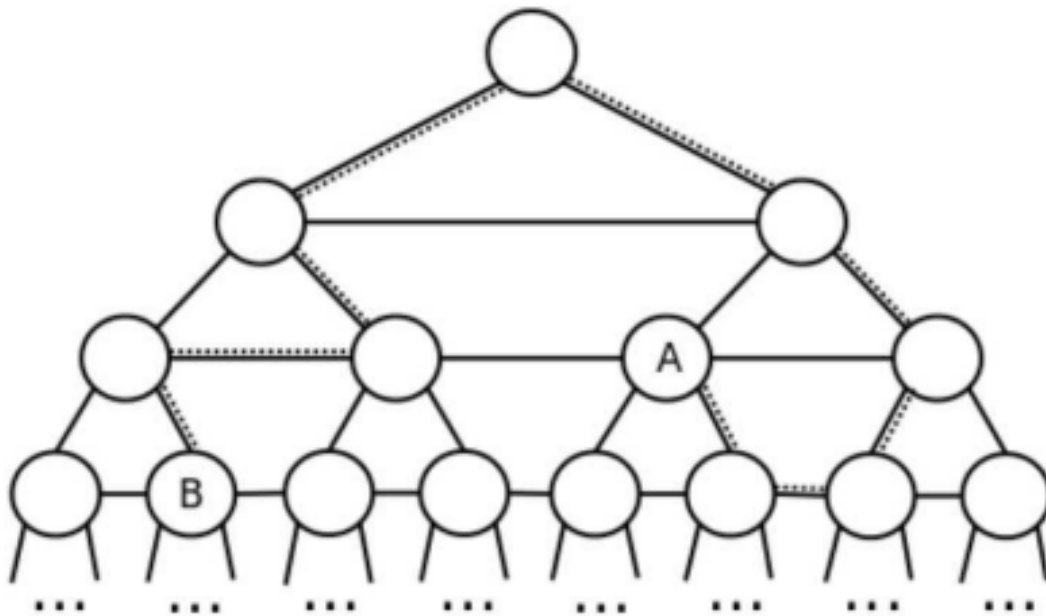
子任务七 (50 分): 无特殊限制。

## 二叉树 (tree)

### 【题目描述】

给出这样一棵“二叉树”：

- 每个节点有左右两个儿子，并如下定义每个节点的高度：假设父亲节点的高度为  $h$ ，那么他的两个儿子的节点的高度都是  $h + 1$ ，相同高度的所有节点称作一层。
- 每个节点的左儿子的子树都在右儿子的子树的左边，每一层相邻的两个节点之间有一条边。下面是一个例子：



每一条从根出发的图上的路径用一个字符串表示，字符串中的每一个字符表示一个移动。字符仅包含如下五种：

- 1：表示移动到当前节点的左儿子
- 2：表示移动到当前节点的右儿子
- U：表示移动到当前节点的父亲节点
- L：表示移动到当前节点同层的左边的节点（保证当前节点在这一层中不是最左边的节点）
- R：表示移动到当前节点同层的右边的节点（保证当前节点在这一层中不是最右边的节点）

用一条路径来表示这条路径的终点，例如路径：221LU 就表示上图中的节点  $A$ 。给出两条路径，你的任务是求出着两条路径的终点之间的最短路。



**【输入格式】**

从文件 *tree.in* 中读入数据。

输入两行，每行一个字符串，分别表示两条路径。

**【输出格式】**

输出到文件 *tree.out* 中。

输出一行，表示两个节点之间的最短路。

**【样例 1 输入】**

```
1 221LU
2 12L2
```

**【样例 1 输出】**

```
1 3
```

**【样例 2 输入】**

```
1 111RRRRRRR
2 222
```

**【样例 2 输出】**

```
1 0
```

**【样例 3 输入】**

```
1 11111
2 222222
```

**【样例 3 输出】**

```
1 10
```

**【样例 4】**

见附加文件中的 *tree/ex\_tree4.in* 与 *tree/ex\_tree4.ans*。

**【数据范围】**

本题采用捆绑测试。

用  $D$  表示所有经过的节点中，深度最大的节点的深度（设根的深度为 0）； $S$  表示输入字符串的最大长度。

对于全部数据， $D \leq 10^5, S \leq 10^5$ 。

子任务一（20 分）： $D \leq 10$ ；

子任务二（20 分）： $D \leq 50$ ；

子任务三（30 分）： $D \leq 10^3$ ；

子任务四（30 分）： $D \leq 10^5, S \leq 10^5$ 。

## 麻将机 (machine)

### 【题目描述】

神奈子是个很爱打麻将的老婆婆，有一天她把她的麻将放进了一个麻将机里准备洗牌。她的麻将很特别，牌面有  $0 \sim 9$ 、 $a \sim z$ 、 $A \sim Z$  共 62 种不同的种类，她把把这些麻将排成一排共  $n$  个，放进了洗牌机里。

因为她很喜欢风，所以她的洗牌机里也有风。这个洗牌机有  $m$  个步骤，第  $i$  个步骤会吹出神奇的风将全部牌面为  $x_i$  的牌吹蚀，使它们的牌面变成  $y_i$ 。麻将机必须执行每种步骤各恰好一次，但是顺序可以由神奈子任意指定。特别的，即使没有牌面为  $x_i$  的牌也可以选择执行第  $i$  个步骤，牌面不会有任何改变，但视为已经完成了该步骤。

因为她很喜欢山，所以她希望洗完牌后麻将的牌面种类尽可能多一点，这样以后堆麻将山的时候会好看点。

她非常好奇，所以问你在合理安排洗牌顺序的情况下，最终不同的麻将牌面种类数量最大是多少？

### 【输入格式】

从文件 *machine.in* 中读入数据。

输入包含多组数据，第一行一个整数  $t$ ，表示数据组数。

对于每组数据，第一行一个字符串  $S$  和一个整数  $m$ ，由空格分隔开。 $S_i$  表示第  $i$  块麻将的牌面，保证所有字符都在  $0 \sim 9$ 、 $a \sim z$ 、 $A \sim Z$  之中， $m$  表示洗牌的总步骤数。

第二行  $m$  个由空格分隔开的长度为 2 的字符串，形如  $x_i y_i$ ，表示第  $i$  个步骤是将牌面为  $x_i$  的牌全部变成  $y_i$ 。保证  $x_i \neq y_i$  且这  $m$  个字符串两两不等。

### 【输出格式】

输出到文件 *machine.out* 中。

对于每组数据，输出一行一个整数表示答案。

### 【样例 1 输入】

```
1 4
2 CODEJAMWORLDFINALS 2
3 AO OY
4 xyz 3
5 xy zx yz
6 CJ 4
7 20 2O HC KS
8 AB 2
9 Ab bA
```

**【样例 1 输出】**

```
1 14
2 2
3 2
4 2
```

**【样例 2】**

见选手下发文件中 machine/ex\_machine2.in 与 machine/ex\_machine2.ans。

**【数据范围】**

对于 30% 的数据，保证每个操作的  $y_i$  互不相等。

对于另外 30% 的数据，保证每个操作的  $x_i$  互不相等。

对于另外 100% 的数据， $t \leq 100, m \leq 61 \times 62, |S| \leq 100$ 。