

牛客 CSP-S 提高组赛前集训营 1

比赛地址: <https://ac.nowcoder.com/acm/contest/1100>

题目名称	仓鼠的石子游戏	乃爱与城市拥挤程度	小 w 的魔术扑克
题目类型	传统型	传统型	传统型
每个测试点 时限	C/C++ 1 秒, 其他语言 2 秒	C/C++ 2 秒, 其他语言 4 秒	C/C++ 1 秒, 其他语言 2 秒
内存限制	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB	C/C++ 256MB, 其他语言 512MB
子任务数目	10	10	10
测试点是否 等分	是	是	是

注意事项

- 所有参与 NOIP 赛前集训营的选手必须遵守约定的纪律:
 - 比赛账号不能外传。
 - 比赛中不能抄袭代码。
 - 比赛中不能恶意卡评测。
 - 报名支付账号即为比赛账号。
 - 一旦报名 NOIP 赛前集训营活动, 不支持退费, 请考虑清楚后报名。
 - 本活动解释权归牛客网所有, 活动介绍未尽事宜以牛客网官方解释为准。
- 欢迎关注“比赛自动姬”公众号, 关注更多比赛资讯~



仓鼠的石子游戏

【题目描述】

仓鼠和兔子被禁止玩电脑，无聊的他们跑到一块空地上，空地上有许多小石子。兔子捡了很多石子，然后将石子摆成 n 个圈，每个圈由 $a[i]$ 个石子组成。然后兔子有两根彩色笔，一支红色一支蓝色。兔子和仓鼠轮流选择一个没有上色的石子涂上颜色，兔子每次可以选择一个还未染色的石子将其染成红色，而仓鼠每次可以选择一个还未染色的石子将其染成蓝色，并且仓鼠和兔子约定，轮流染色的过程中不能出现相邻石子同色，谁不能操作他就输了。假设他们两个都使用了最优策略来玩这个游戏，并且兔子先手，最终谁会赢得游戏？。

【输入格式】

第一行输入一个正整数 T ，表示有 T 组测试案例。

每组测试案例的第一行输入一个 n ，表示有 n 圈石子。第二行输入 n 个正整数 $a[i]$ ，表示每个圈的石子数量。

【输出格式】

对于每组测试案例，如果兔子赢了，输出"rabbit"(不含引号)如果仓鼠赢了，输出"hamster"(不含引号)。

【样例 1 输入】

4

1

3

1

1

2

1 3

3

999 1000 1000000000

【样例 1 输出】

hamster

rabbit

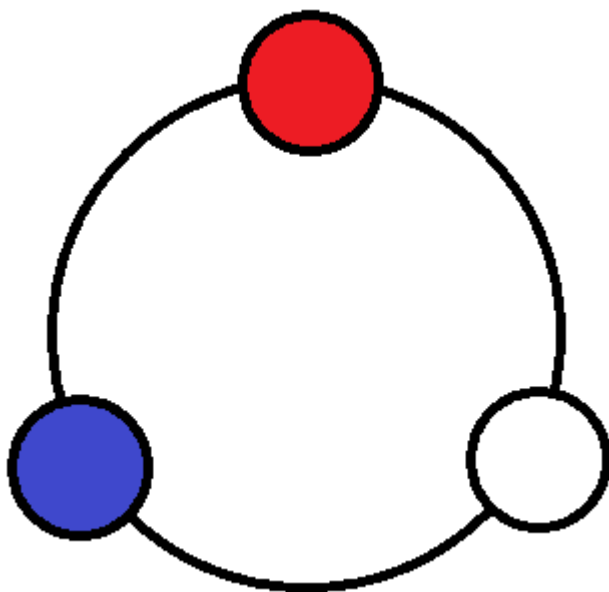
rabbit

hamster

【样例 1 解释】

对于第一组案例：只有 1 圈石子，并且石圈的大小为 3。

兔子先手，随便找了一个石子染成红色，接下来仓鼠后手找一个未染色的石子染成蓝色，此时结果如下图所示。



如果兔子将最后一个石子染成红色，这将导致相邻石子同色，根据规则，他输掉了比赛，所以仓鼠获得了最终的胜利。

对于第二组案例：只有 1 圈石子，并且石圈的大小为 1。

兔子先手，将唯一的一个石子染成了红色，接下来由于没有未着色的石子，所以仓鼠由于无法操作而输掉了比赛，兔子取得了最终的胜利。

对于第三组案例：有两个石圈，大小分别为 1,3，兔子首先将大小为 1 的石圈中唯一一个石子染成了红色，接下来仓鼠由于类似第一组案例中的原因输掉比赛，兔子取得了最终的胜利。

【数据范围】

对于前 30%的数据，满足 $n = 1, 1 \leq a[i] \leq 7, 1 \leq T \leq 10$ 。

对于前 60%的数据，满足 $1 \leq n \leq 10^3, 1 \leq a[i] \leq 7, 1 \leq T \leq 10^2$ 。

对于前 100%的数据，满足 $1 \leq n \leq 10^3, 1 \leq a[i] \leq 10^9, 1 \leq T \leq 10^2$ 。

对于测试点 6，在满足前 60%的数据条件下，额外满足 $a[i] = 1$ 。

乃爱与城市拥挤程度

【题目描述】

乃爱天下第一可爱！

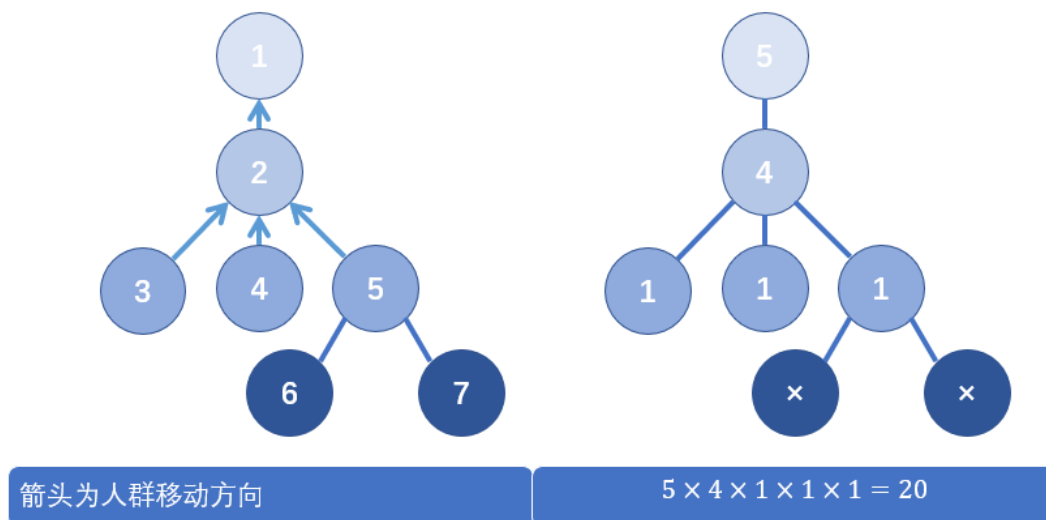
乃爱居住的国家有 n 座城市，这些城市与城市之间有 $n-1$ 条公路相连接，并且保证这些城市两两之间直接或者间接相连。

我们定义两座城市之间的距离为这两座城市之间唯一简单路径上公路的总条数。

当乃爱位于第 x 座城市时，距离城市 x 距离不大于 k 的城市中的人都会认为乃爱天下第一可爱！

认为乃爱天下第一可爱的人们决定到乃爱所在的城市去拜访可爱的乃爱。我们定义这些城市的拥挤程度为：

距离城市 x 距离不大于 k 的城市中的人到达城市 x 时经过该城市的次数。例如：



假设 $k=2$ ，乃爱所在的城市是 1 号城市，树结构如上图所示时，受到影响的城市为 1,2,3,4,5，因为五个城市距离 1 号城市的距离分别为：0,1,2,2,2，所以这五个城市都会认为乃爱天下第一。

1 号城市到 1 号城市经过了 1 号城市。

2 号城市到 1 号城市经过了 1 号、2 号城市。

3 号城市到 1 号城市经过了 1 号、2 号、3 号城市。

4 号城市到 1 号城市经过了 1 号、2 号、4 号城市。

5 号城市到 1 号城市经过了 1 号、2 号、5 号城市。

所以 1 号城市的拥挤程度是 5，2 号城市的拥挤程度是 4，3 号、4 号、5 号城市的拥挤程度都是 1。

现在小 w 想要问你当乃爱依次位于第 1、2、3、4、5...n 座城市时，有多少座城市中的人认为乃爱天下第一，以及受到影响城市的拥挤程度的乘积，由于这个数字会很大，所以要求你输出认为乃爱天下第一的城市拥挤程度乘积 $\text{mod } 10^9 + 7$ 后的结果。

【输入格式】

第一行是两个正整数 n, k 表示城市数目，以及距离乃爱所在城市距离不大于 k 的城市中的人认为乃爱天下第一！

接下来 $n - 1$ 行，每行两个正整数 u, v ，表示树上一条连接两个节点的边。

【输出格式】

输出两行。

第一行 n 个整数，表示当乃爱依次位于第 1、2、3、4、5...n 座城市时，有多少座城市中的人认为乃爱天下第一。

第二行 n 个整数，表示当乃爱依次位于第 1、2、3、4、5...n 座城市时，受影响的拥挤程度乘积 $\text{mod } 10^9 + 7$ 后的结果。

【样例 1 输入】

7 2

1 2

2 3

2 4

2 5

5 6

5 7

【样例 1 输出】

5 7 5 5 7 4 4

20 21 20 20 28 12 12

【数据范围】

对于前 10%的测试点满足 $1 \leq n \leq 10, 1 \leq k \leq 10$ ，树结构随机生成。

对于前 30%的测试点满足 $1 \leq n \leq 10^3, 1 \leq k \leq 10$ ，树结构随机生成。

对于前 70%的测试点满足 $1 \leq n \leq 10^5, 1 \leq k \leq 10$ ，树结构随机生成。

对于前 100%的测试点满足 $1 \leq n \leq 10^5, 1 \leq k \leq 10$ ，树结构为手动构造。

对于测试点 4，在满足其前 70%的测试点条件下，额外满足 $k = 1$ 。

对于测试点 5，在满足其前 70%的测试点条件下，额外满足 $k = 2$ 。

对于测试点 10，在满足其前 100%的测试点条件下，额外满足树结构退化成一个链。

小 w 的魔术扑克

【题目描述】

小 w 喜欢打牌，某天小 w 与 dogenya 在一起玩扑克牌，这种扑克牌的面值都在 1 到 n ，原本扑克牌只有一面，而小 w 手中的扑克牌是双面的魔术扑克（正反两面均有数字，可以随时进行切换），小 w 这个人就准备用它来出老千作弊。小 w 想要打出一些顺子，我们定义打出一个 l 到 r 的顺子需要面值为从 l 到 r 的卡牌各一张。小 w 想问问你，他能否利用手中的魔术卡牌打出这些顺子呢？

【输入格式】

首先输入一行 2 个正整数 n, k ，表示牌面为 $1 \sim n$ ，小 w 手中有 k 张魔术扑克牌。
然后输入 k 行，每行两个数字，表示卡牌的正面和反面的面值。
接下来输入一行一个正整数 q ，表示 q 组查询，然后每组占一行查询输入两个整数 l, r 。表示查询小 w 能否打出这么一个 l 到 r 的顺子。

【输出格式】

对于输出 "Yes" 表示可以，"No" 表示不可以。（不含引号）

每个查询都是独立的，查询之间互不影响。

【样例 1 输入】

5 3

1 2

2 3

4 4

3

1 2

2 4

1 4

【样例 1 输出】

Yes

Yes

No

【样例 1 解释】

对于顺子 1~2，可以选择第一张卡牌作为'1'使用，选择第二张卡牌作为'2'使用。

对于顺子 2~4，可以选择第一张卡牌作为'2'使用，选择第二张卡牌作为'3'使用，
选择第三张卡牌作为'4'使用。

对于顺子 1~4，由于牌的数目都不够，显然无法打出。

【样例 2 输入】

4 3

1 1

2 2

4 4

3

1 2

1 4

4 4

【样例 2 输出】

Yes

No

Yes

【样例 1 解释】

该样例具有测试点 4 的特殊性质。

【数据范围】

对于前 10%的测试点，保证 $1 \leq n \leq 10, 1 \leq k \leq 10, 1 \leq q \leq 10, 1 \leq l \leq r \leq n$ 。

对于前 20%的测试点，保证 $1 \leq n \leq 11, 1 \leq k \leq 10, 1 \leq q \leq 100, 1 \leq l \leq r \leq n$ 。

对于前 30%的测试点，保证 $1 \leq n \leq 50, 1 \leq k \leq 50, 1 \leq q \leq 500, 1 \leq l \leq r \leq n$ 。

对于前 100%的测试点，保证 $1 \leq n \leq 105, 1 \leq k \leq 105, 1 \leq q \leq 105, 1 \leq l \leq r \leq n$ 。

对于测试点 4，在满足前 100%的测试点条件下，额外保证所有卡牌正面上的数字等于其反面上的数字，但不同扑克牌上的数字不保证相同。