西默夫（Simurgh）

根據沙納瑪（Shahnameh）中的古代波斯傳說，Zal，傳奇的波斯英雄，瘋狂地愛上了 Kabul 王國的公主Rudaba。在 Zal 向 Rudaba 求婚時，Rudaba 的父親給他了一個挑戰。

在波斯有 n 個城市，標記為從 0 到 $n-1$，以及 m 條雙向道路，標記為從 0 到 $m-1$。 每條道路連接兩個不同的城市。每一對城市至多會被一條道路連接。有些道路是禦道（royal roads），專用於皇室行駛，但這是保密的。Zal 的任務是找出哪些道路是禦道。

Zal 有一張包括所有城市和所有道路的波斯地圖。他不知道哪些道路是禦道，但是他可以求救於Simurgh——好心的神鳥、Zal 的保護者。然而，Simurgh 並不想直接告訴他哪些道路是禦道。取而代之，Simurgh告訴Zal，所有禦道的集合是一個黃金集合（golden set）。一個道路的集合是黃金集合，當且僅當：

- 它恰好包含 $n-1$ 條道路，而且
- 對於每一對城市，僅沿著這個集合中的道路即可從其中一個城市抵達另外一個城市。

此外，Zal 可以問 Simurgh 一些問題。對於每個問題：

1. Zal 選出道路的一個黃金集合，然後
2. Simurgh 會告訴 Zal，在所選擇的黃金集合中有多少條道路是禦道。

你的程式可以問 Simurgh 最多 q 個問題，以此幫助 Zal 找出禦道的集合。評測工具將扮演 Simurgh 的角色。

實現細節

你需要實現下面的函數：

```
int[] find_roads(int n, int[] u, int[] v)
```

- n：城市的數量，
- u 和 v：均為長度為 m 的陣列。對於所有 $0 \leq i \leq m-1$，$u[i]$ 和 $v[i]$ 是被道路 i 所連接的城市。
- 該函數需要返回一個長度為 $n-1$ 的陣列，其中包括了所有禦道的標號（可以以任意的順序給出）。

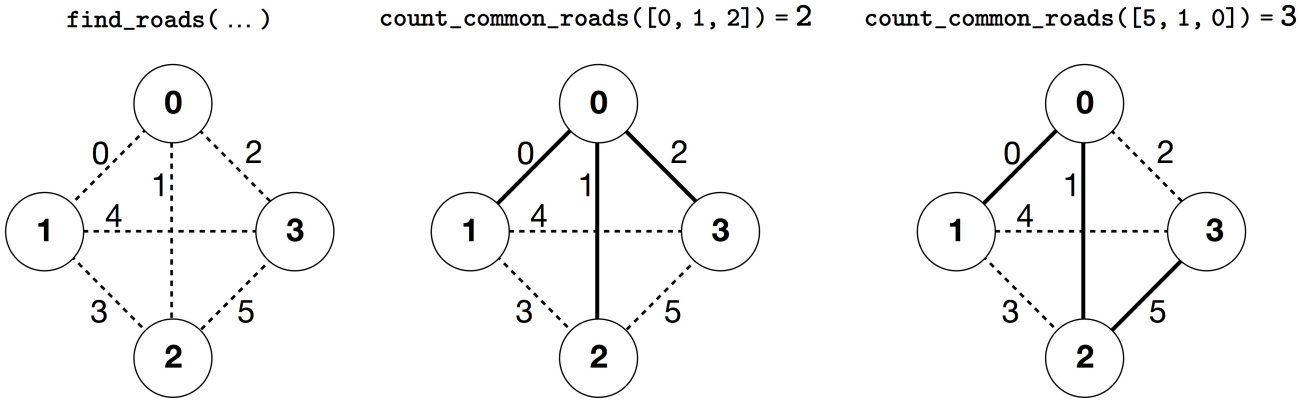
你的程式至多只能調用評測工具中的如下函數 q 次：

```
int count_common_roads(int[] r)
```

- r：長度為 $n-1$ 的陣列，其中包括了一個黃金集合中的道路標號（可以以任意的順序給出）。
- 該函數將返回 r 中的禦道數量。

例子

```
find_roads(4, [0, 0, 0, 1, 1, 2], [1, 2, 3, 2, 3, 3])
```



這個例子中有 4 個城市和6條道路。 我們將連接城市 a 和 b 的道路表示為 (a, b)。這些道路按照下面的順序被標為從 0 到 5：$(0,1)$，$(0,2)$，$(0,3)$，$(1,2)$，$(1,3)$ 和 $(2,3)$。每個黃金集合包含 $n-1=3$ 條道路。

假設禦道是標號為 0，1 和 5 的道路，即 $(0,1)$，$(0,2)$ 和 $(2,3)$。這樣的話：

- `count_common_roads([0, 1, 2])` 返回 2。該詢問涉及到標號為 $0, 1$ 和 2 的道路，即 $(0,1)$， $(0,2)$ 和 $(0,3)$。其中有兩條道路是禦道。
- `count_common_roads([5, 1, 0])` 返回 3。該詢問涉及到所有的禦道。

函數 `find_roads` 需要返回 `[5, 1, 0]` 或任意其他包含這三個元素且長度為 3 的陣列。

注意，下面列出的調用是不允許的：

- `count_common_roads([0, 1])`：這裡 r 的長度不是 3。
- `count_common_roads([0, 1, 3])`：這裡 r 不是一個黃金集合，因為無法僅沿道路 $(0,1)$，$(0,2)$，$(1,2)$ 就從城市 0 走到城市 3。

限制條件

- $2 \le n \le 500$
- $n-1 \le m \le n(n-1)/2$
- $0 \le u[i], v[i] \le n-1$（對於所有 $0 \le i \le m-1$）
- 對於所有 $0 \le i \le m-1$，道路 i 連接兩個不同的城市（即 $u[i] \neq v[i]$）。
- 每對城市之間至多連有一條道路。
- 經由這些道路，可以在任意一對城市之間來往。

- 所有的禦道組成一個黃金集合。
- `find_roads` 可以調用 `count_common_roads` 最多 q 次。在每次調用中，由 r 所給出的道路必須是一個黃金集合。

子任務

1. （13分）$n \leq 7$，$q = 30\,000$
2. （17分）$n \leq 50$，$q = 30\,000$
3. （21分）$n \leq 240$，$q = 30\,000$
4. （19分）$q = 12\,000$，在任意兩個城市之間都連有一條道路
5. （30分） $q = 8000$

評測工具示例

評測工具示例將讀入下述格式的輸入資料：

- 第 1 行： n m
- 第 $2 + i$ 行（對於所有 $0 \leq i \leq m - 1$）： $u[i]$ $v[i]$
- 第 $2 + m$ 行： $s[0]$ $s[1]$ $\ldots$ $s[n - 2]$

這裡$s[0], s[1], \ldots, s[n - 2]$ 是所有禦道的標號。

如果 `find_roads` 最多調用 `count_common_roads` 了 $30\,000$ 次，而且正確地返回了禦道的集合，評測工具示例將會輸出 `YES`。 否則評測工具示例將會輸出 `NO`。

需要明確的是，評測工具示例中的函數 `count_common_roads` 不會檢查 r 是否滿足一個黃金集合的所有條件。相對應地，它會對陣列r中的禦道進行計數，並且返回。然而，在你提交的程式調用 `count_common_roads` 時，如果傳給它的不是對應某個黃金集合的標號集合，評測結果將會是 `'Wrong Answer'`.

技術提示

出於效率方面的考慮，面向 C++ 和 Pascal 的函數 `count_common_roads` 使用了傳引用調用（call by reference）的方式。你可以與平常一樣調用這個函數。評測工具確保不會改變 r 中的值。