

{% note info %} **摘要** Title: 1471. 牛奶工厂 Tag: 图的遍历、DFS、floyd传递闭包 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

1471. 牛奶工厂

• 题意

牛奶生意正红红火火！农夫约翰的牛奶加工厂内有 N 个加工站，编号为 $1 \dots N$ ，以及 $N-1$ 条通道，每条连接某两个加工站。（通道建设很昂贵，所以约翰选择使用了最小数量的通道，使得从每个加工站出发都可以到达所有其他加工站）。为了创新和提升效率，约翰在每条通道上安装了传送带。不幸的是，当他意识到传送带是单向的已经太晚了，现在每条通道只能沿着一个方向通行了！所以现在的情况不再是从每个加工站出发都能够到达其他加工站了。然而，约翰认为事情可能还不算完全失败，只要至少还存在一个加工站 i 满足从其他每个加工站出发都可以到达加工站 i 。注意从其他任意一个加工站 j 前往加工站 i 可能会经过 i 和 j 之间的一些中间站点。请帮助约翰求出是否存在这样的加工站 i 。

• 思路

◦ DFS爆搜 $O(n^2)$

能被所有点走到的点一定是出度为0的点，在建图时统计这些点，之后对其余点进行DFS，看是否每个点都能到这些点，输出最小的即可

◦ floyd传递闭包 $O(n^3)$

求出每两个点是否互通，之后枚举每个点即可

◦ 结论 $O(n)$

- 若有解，则解必定唯一
- 出度为0的点有且仅有一个是有解的充分必要条件

则只需统计出度为0的点即可，如果只有一个，那么就是答案

• 代码

◦ DFS爆搜 1224 ms

```
...  
Author: NEFU AB-IN  
Date: 2022-02-11 09:35:06  
FilePath: \ACM\Acwing\1471.py
```

```

LastEditTime: 2022-02-11 09:56:24
'''
import sys

sys.setrecursionlimit(10000000)

from collections import Counter

N = 150

g = [[] for _ in range(N)]
deg = [0] * N #出度
a = [] #出度为0的点
b = Counter() #统计每个点能被到几次

def dfs(x):
    for y in g[x]:
        b[y] += 1
        dfs(y)

if __name__ == "__main__":
    n = int(input())
    for i in range(n - 1):
        x, y = map(int, input().split())
        g[x].append(y)
        deg[x] += 1
    for i in range(1, n + 1): #枚举每个点，看哪个度为0
        if deg[i] == 0:
            a.append(i)
    for i in range(1, n + 1): #dfs每个点
        dfs(i)
    for i in a:
        if b[i] == n - 1:
            print(i)
            exit(0)
    print(-1)

```

◦ floyd传递闭包 871 ms

- 枚举点的时候，**一定要看清楚点的范围**，是从0还是从1开始的
- **别忘了初始化**，0代表两点不通，1代表两点相通

```

'''
Author: NEFU AB-IN
Date: 2022-02-12 14:10:48
FilePath: \ACM\Acwing\1471.1.py
LastEditTime: 2022-02-12 14:17:58
'''

N = 155

```

```

g = [[0 for _ in range(N)] for _ in range(N)]

if __name__ == "__main__":
    n = int(input())
    for i in range(n - 1):
        x, y = map(int, input().split())
        g[x][y] = 1
    for i in range(1, n + 1): #初始化floyd, 每个点都能通向自己
        g[i][i] = 1
    for k in range(1, n + 1):
        for i in range(1, n + 1):
            for j in range(1, n + 1):
                if g[i][k] and g[k][j]:
                    g[i][j] = 1 #代表i与j相通
    for i in range(1, n + 1):
        cnt = 0
        for j in range(1, n + 1):
            if g[j][i]:
                cnt += 1
        if cnt == n:
            print(i)
            exit(0)
    print(-1)

```

◦ **结论** 739ms

```

'''
Author: NEFU AB-IN
Date: 2022-02-12 14:36:01
FilePath: \ACM\Acwing\1471.2.py
LastEditTime: 2022-02-12 14:38:12
'''

N = 155
deg = [0] * N

if __name__ == "__main__":
    n = int(input())

    for i in range(n - 1):
        x, y = map(int, input().split())
        deg[x] += 1

    cnt = 0
    res = 0
    for i in range(1, n + 1):
        if deg[i] == 0:
            cnt += 1
            res = i
    if cnt == 1:
        print(res)

```

```
else:  
    print(-1)
```