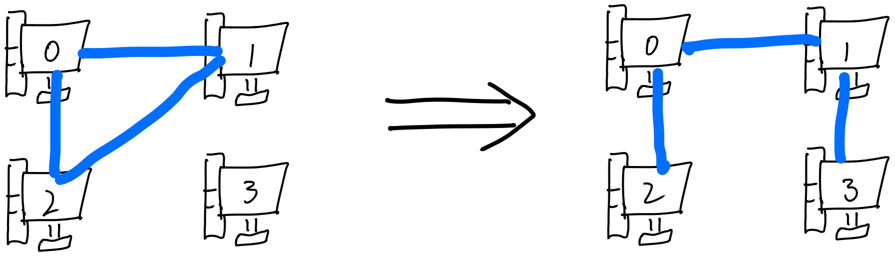
There are n computers numbered from 0 to n-1 connected by cables connections forming a network where connections[i] = [a, b] represents a connection between computers a and b. Any computer can reach other computer directly or indirectly through the network.

Given an initial network connections. You can extract certain cables between two directly connected computers, and place them between any pair of disconnected computers to make them directly connected. Return the minimum number of times you need to do this in order to make all the computers connected. If it's not possible, return -1.

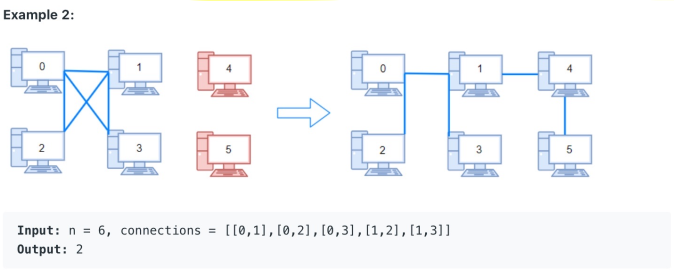
Example 1:



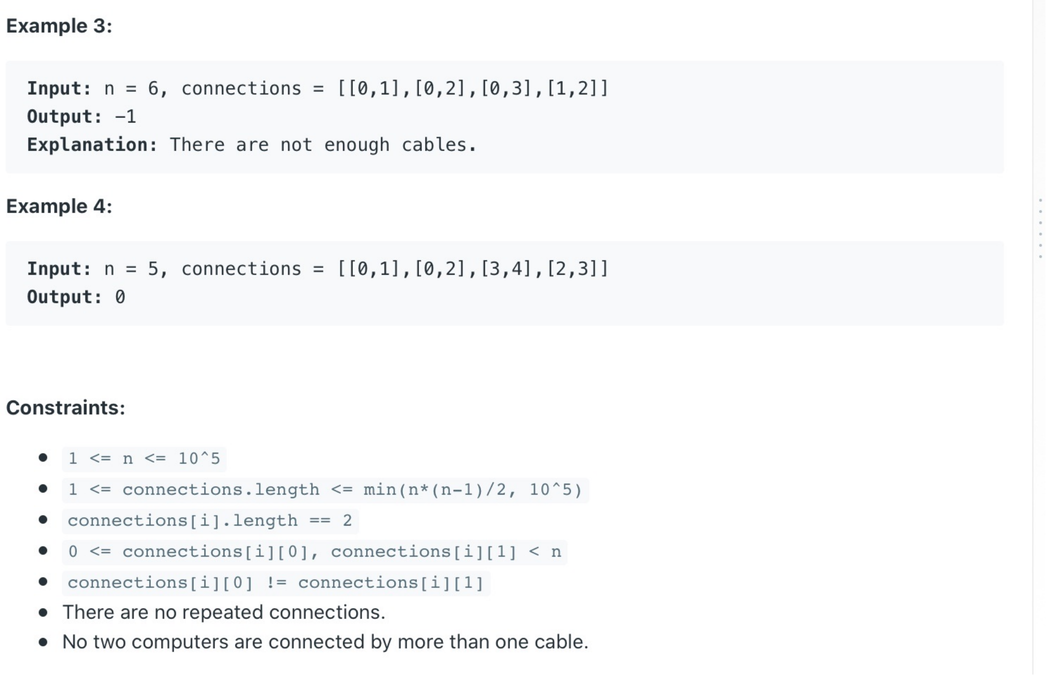
Input: n=4, connections = [[0,1], [0,2], [1,2]]

Output: 1

Explanation: Remove cable between computer 1 and 2 and place between computers 1 and 3


Example 2:

```
Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2],[1,3]]
Output: 2
```



Example 3:

```
Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2]]
Output: -1
Explanation: There are not enough cables.
```

Example 4:

```
Input: n = 5, connections = [[0,1],[0,2],[3,4],[2,3]]
Output: 0
```

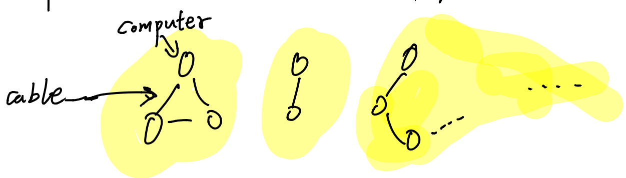
Constraints:

- `1 <= n <= 10^5`
- `1 <= connections.length <= min(n*(n-1)/2, 10^5)`
- `connections[i].length == 2`
- `0 <= connections[i][0], connections[i][1] < n`
- `connections[i][0] != connections[i][1]`
- There are no repeated connections.
- No two computers are connected by more than one cable.

分析：① 首先，n个计算机，m个电缆，只有当 m >= n-1 时有解.

② 在有解的情况下，

Computer 和 cable 的分布应该是这样的：



会有多个子网络，只要把这些子网连成一个网络即可.

n个子网络，只要 n-1 条 cable 即可连成一个网络.

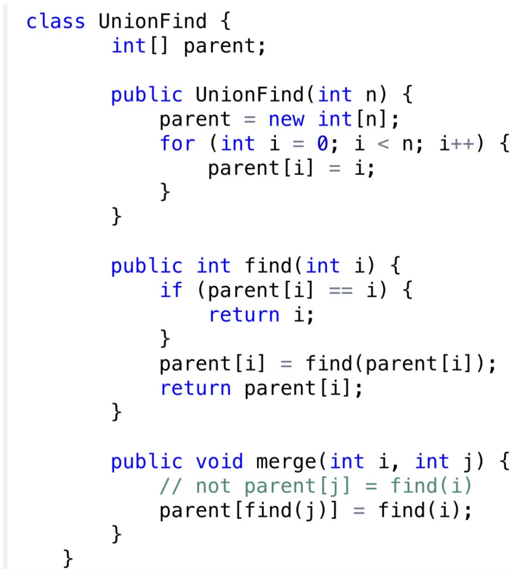
所以，这个问题可以转化为计算子网络的个数

本题的解法需要用到并查集.

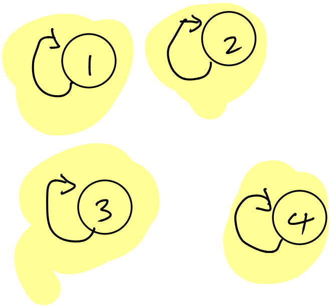
并查集主要用于解决元素分组问题. 它管理一系列不相交的集合，并支持两种操作：

① 合并(Union)：把两个不相交的集合合并为一个集合.

② 查询(Find)：查询一个元素所在的集合.

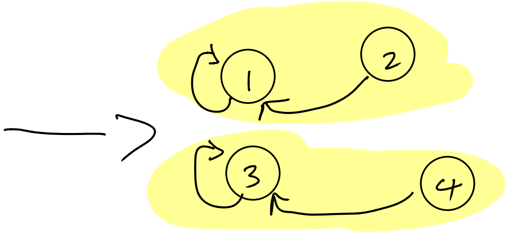


```
class UnionFind {
        int[] parent;

        public UnionFind(int n) {
            parent = new int[n];
            for (int i = 0; i < n; i++) {
                parent[i] = i;
            }
        }

        public int find(int i) {
            if (parent[i] == i) {
                return i;
            }
            parent[i] = find(parent[i]);
            return parent[i];
        }

        public void merge(int i, int j) {
            // not parent[j] = find(i)
            parent[find(j)] = find(i);
        }
    }
```



上面是一个并查集结构. 初始每个元素是一个集合，它们的根节点是自己.

合并： ②并入①
③并入④.



{③,④} 并入{①,②}