

684. 冗余连接

算法 图 并查集

题目: Redundant Connection

语言: python3

英文版链接: <https://leetcode.com/problems/redundant-connection/description/>

中文版链接: <https://leetcode-cn.com/problems/redundant-connection/>

题目分析

本题是并查集的一个简单应用，题中说只有一条附加边，因此当出现第一个不能union的边即为附加边。

如果对并查集不太熟悉的话，可以参考下面连个链接：

[并查集的直观理解](#)

[并查集具体原理和应用](#)

答案

实现一个并查集的类，并实现其三个主要的操作。

```
class UnionFind:
    def __init__(self, n):
        self.ids = []
        for i in range(n + 1):
            self.ids.append(i)

    def union(self, u, v):
        u_id = self.find(u)
        v_id = self.find(v)
        if u_id == v_id:
            return
        for i in range(len(self.ids)):
            if self.ids[i] == u_id:
                self.ids[i] = v_id

    def find(self, p):
        return self.ids[p]

    def connect(self, u, v):
        return self.find(u) == self.find(v)
```

```
class Solution:
    def findRedundantConnection(self, edges):
        uf = UnionFind(len(edges))
        for e in edges:
            u, v = e[0], e[1]
            if uf.connect(u, v):
                return u, v
            uf.union(u, v)

        return -1, -1
```