

Union Find

#G面试准备

Number of Islands

Number of Islands - LeetCode

- 写出union find类
- 用到计数 有多少个集合的count
- 先数1的个数 初始化count
- 再对每个1上下左右找1
- 上下左右不仅要满足边界条件 还要是1 不要忘记判断
- 二维转一维是 $x * m + y$, 不是 n ; n 是行数 m 是列数

Number of Islands II

Number of Islands II

WA test case:

```
3
3
[[0,1],[1,2],[2,1],[1,0],[0,2],[0,0],[1,1]]
```

错误原因:

```
}
    public void connect(int x, int y) {
        int root_x = find(x);
        int root_y = find(y);
        if (root_x != root_y) {
            father[x] = root_y;
            count--;
        }
    }
    public int getCount() {
```

- $\text{father}[\text{root_x}] = \text{root_y}$ 才对
- 如果只改了 $\text{father}[x]$ 并没有根本上改全部的 father (其实还是不是很理解 但是一定要记住 要改

的是root的father)

- 还是一个跟上面一样的Union Find问题
- 只不过这回要在每次加入新的点的时候 改count+1
- 然后再上下左右去找是1的点connect connect里面如果不是同一个集合会减1

684. Redundant Connection

Redundant Connection - LeetCode

- 看起来是一道图里找树bad edge的题
- 其实可以用union find做
- 给的是edges数组 遍历edges数组 如果两个节点不是同一个集合 就连接
- 是的话直接输出
- 注意输入的是正好比树多一条边 所以可以初始化unionfind有边数n个entry
- 又因为是从1..n 可以多加一位 把0忽略 比较方便使用