Leetcode 130. 被围绕的区域

宋星霖

2024 年 2 月 20 日

1 题目描述

1.1 题目描述

给你一个 m x n 的矩阵 board ，由若干字符 'X' 和 'O' ，找到所有被 'X' 围绕的区域，并将这些区域里所有的 'O' 用 'X' 填充。

1.2 示例 1

输入:board = [["X","X","X","X"],["X","O","O","X"],["X","X","O","X"],["X","O","X","
输出:[["X","X","X","X"],["X","X","X","X"],["X","X","X","X"],["X","O","X","X"]]
解释：被围绕的区间不会存在于边界上，换句话说，任何边界上的 'O' 都不会被填充为 'X'。任何不在边界上，或不与边界上的 'O' 相连的 'O' 最终都会被填充为 'X'。如果两个元素在水平或垂直方向相邻，则称它们是“相连”的。

1.3 示例 2

输入：board = [["X"]] 输出：[["X"]]

2 算法分析

2.1 第一步：构建并查集

由于并查集在 Leetcode 128 文章讲过，在此不再赘述。

2.2 第 2 步：将 2 维映射到 1 维

由于并查集只能处理一维数据，所以需要进行映射。$board[i][j]$ 的下标为 $i \times n + j + 1$ 。

2.3 第 3 步：合并

遍历数组。如果当前位置为 'O' 并且它的右边或下面的位置是 'O'，则合并这两个节点。

还有一种特殊情况：当前遍历的位置在边上，这时候，将当前位置与 0 连接。前面计算下标加一个 1 的原因就是因为要预留出 0 位置。

2.4 第 4 步：替换

遍历数组。如果当前位置为 'O' 并且当前位置不与 0 位置相连，则将当前位置变为 'X'。

3 代码演示

Listing 1: 代码

```
class UnionSet {
    public :
    UnionSet(int n) : fa(n + 1) {
        for(int i = 0; i <= n; i++) fa[i] = i;
    }
    int find(int x) {
        return fa[x] = (fa[x] == x ? x : find(fa[x]));
    }
    void merge(int a, int b) {
        if(find(a) == find(b)) return;
        fa[find(a)] = find(b);
    }
    vector<int> fa;
};

class Solution {
    public:
    void solve(vector<vector<char>>& board) {
        int m = board.size(), n = board[0].size();
        UnionSet u(m * n);
        for(int i = 0;i < m;i++) {
            for(int j = 0;j < n;++j) {
                int ind = i * n + j + 1;
                if(board[i][j] != 'O') continue;
                if(i == 0 || i == m - 1) u.merge(ind,0);
                if(j == 0 || j == n - 1) u.merge(ind,0);
                if(j + 1 < n && board[i][j + 1] == 'O') u.merge(ind,ind + 1);
                if(i + 1 < m && board[i + 1][j] == 'O') u.merge(ind,ind + n);
            }
        }
        for(int i = 0;i < m;++i) {
            for(int j = 0;j < n;++j) {
                if(board[i][j] != 'O') continue;
                int ind = i * n + j + 1;
                if(u.find(ind) != u.find(0)) board[i][j] = 'X';
            }
        }
    }
};
```