Disjoint_Sets.cpp

Saved

```cpp
#include <bits/stdc++.h>
using namespace std;

class UnionFind {
    vector<vector<pair<int, int>>> root;

  public:
    int components;

    UnionFind(int n, int m) {
        root = vector<vector<pair<int, int>>>(n, vector<pair<int, int>>(m));
        for (int i = 0; i < n; i++)
            for (int j = 0; j < m; j++)
                root[i][j] = {i, j};
        components = n * m;
    }

    pair<int, int> find(int i, int j) {
        if (make_pair(i, j) == root[i][j]) return {i, j};
        return root[i][j] = find(root[i][j].first, root[i][j].second);
    }

    void join(int i, int j, int x, int y) {
        pair<int, int> r1 = find(i, j), r2 = find(x, y);
        if (r1 == r2) return;
        root[r2.first][r2.second] = r1;
        components--;
    }
};

int numIslands(vector<vector<int>>& grid) {
    int n = grid.size();
    if (!n) return 0;
    int m = grid[0].size();
    UnionFind dsu(n, m);
    int di[4] = {1, -1, 0, 0}, dj[4] = {0, 0, 1, -1};

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (grid[i][j] == 0) {
                dsu.components--;
                continue;
            }
            for (int k = 0; k < 4; k++) {
                if (i + di[k] >= n || i + di[k] < 0 || j + dj[k] >= m || j + dj[k] < 0) continue;
                if (grid[i + di[k]][j + dj[k]] == 0) continue;
                dsu.join(i, j, i + di[k], j + dj[k]);
            }
        }
    }

    return dsu.components;
}

int main() {
    cout << "Number of Rows: ";
    int n;
    cin >> n;
    cout << "Number of Cols: ";
    int m;
    cin >> m;
    cout << "Enter the Grid:\n";
    vector<vector<int>> grid(n, vector<int>(m));
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> grid[i][j];

    cout << "Number of Islands: " << numIslands(grid) << endl;

    return 0;
}
```

Terminal

```
Number of Rows: 6
Number of Cols: 6
Enter the Grid:
1 0 1 1 0 1
1 1 0 0 0 1
0 0 1 1 0 0
1 1 0 0 1 1
1 1 1 0 0 1
0 0 1 0 1 1
Number of Islands: 6
```