07-10-2020

B Pranceeth

1BM18CS023

# ADS-lab

Find number of islands using disjoint sets.

```
// Class to represent disjoint set data structure (also contains functions to perform union, find parent element, and make sets)
class DUS {
    vector<int> rank, parent;
    void makeSet() { }
    int find(int x) { }
    void Union(int a, int b) { }
};

// function to count the number of islands
int countIslands(vector<vector<int>> a)
    DUS *dus = new DUS(n*m); // object of DUS class of size = n*m
// traverse through all elements in matrix
for j -> 0 to n
    for k -> 0 to n
        if (a[j][k] == 0) // If element is 0 then it is not part of any island
            continue;
// else check all 8 neighbours. Do union if neighbours is also 1
// checking right neighbour
if (k+1 < m && a[j][k+1] == 1)
    dus -> Union(j*(m)+k, j*m+k+1);
```

B Pranceeth

```
        // checking neighbour below
        if (j+1 < n && a[j+1][k] == 1)
            dus -> Union(j*m + k, (j+1)*m + k);
        // similarly check for other 6 neighbours
    }
}
// Array to store frequency of each set
int *c = new int[n*m];
int numberOfIslands = 0;
for (int j=0; j<n; j++){
    for(int k=0; k<m; k++)
    {
        if(a[j][k] == 1{
            int x = dus->find(j*m+k);
            if (c[x] == 0){
                numberOfIslands++;
                c[x]++;
            }
            else
                c[x]++;
        }
    }
}
return numberOfIslands;
}
```

B. Pranesh