ADS LAB-4

Shikha.N
1BM18CS149
7-10-20

Finding the number of islands using disjoint sets:

pseudo code

```
class Disjoint {
    int n; int count.
    vector<int> parent;
public Disjoint (int n) {
        parent[n];
        for (int i=0; i<n; i++) {
            parent[i] = i;
        } // all elements are in their own set.
        count = 0
    }
public int find (int x) {
        if (parent[x] != x) {
            return find (parent[x]);
        }
        return x;
    }
// connecting the sets that includes x & y respectively.
public void join (int x, int y) {
    int Rootx = find(x);
    int Rooty = find(y);
    if (Rootx != Rooty) {
        parent[Rootx] = Rooty;
        count--;
    }
}
```


Shikha

```
public void Counter(int n){
    count = n;
}
public int count(){
    return count;
}
};
```


Chithan

```
int countIslands(vector<vector<int>> m)
{
    int count = 0;
    int a = m.size();
    int b = m[0].size();
    for(int i=0; i<a; i++)
    {
        for(int j=0; j<b; j++)
        {
            if((a[i][j]) == 1)
            {
                check all 8 neighbours
                & if value is 1, take the
union of the index and its neighbour
            }
        }
    }
    Disjoint d = new Disjoint(a*b);
    for(int i=0; i<a; i++){
        for(int j=0; j<b; j++){
            if(m[i][j])
                count++;
        }
    }
    // set d.counter(count);
    return d.count();
}
```