Name: Rohit Kudache
USN: 1BM18CS083 ('B' Sec)

Page No.
Date

Lab-3

Number of Island using Disjoint Set

Algorithm:

1) Convert m x n (m-rows n-columns) matrix (matrix [][]) to 1-D array (parent [] of length m x n
   for each mat [i] [j], match (i, j) to (n x i + j) so index (n x i + j) represents mat[i][j] parent [n x i + j] represents which subset the mat [i] [j] belongs to.

2) Count all islands.

3) loop through the matrix - 2D (mat [][]) if find an island X (point to root parent element S), check adjacent neighbours if any adjacent island present, it should be in same subject of X.
   If there is an adjacent island Y and is not in the same subset of X.

4) while one island is merged to a subset, the number of island will be decremented by 1, after we write all the connected islands, we get the number of island.

Rohit

code:

```
int countIslands(vector<vector<int>> a){
    int n = a.size();
    int m = a[0].size();
DisjointUnionSets *dus = new DisjointUnionSets(n*m);
    for(int j=0; j<n; j++){
        for(int k=0; k<m; k++){
            if(a[j][k]==0)
                continue;
            if(j+1<n && a[j+1][k]==1)
                dus->union(j*(m)+k, (j+1)*(m)+k);
            if(j-1>=0 && a[j-1][k]==1)
                dus->union(j*(m)+k, (j-1)*(m)+k);
            if(k+1<m && a[j][k+1]==1)
                dus->union(j*(m)+k, (j)*(m)+k+1);
            if(k-1>=0 && a[j][k-1]==1)
                dus->union(j*(m)+k, (j)*(m)+k-1);
            if(j+1<n && k+1<m && a[j+1][k+1]==1)
                dus->union(j*(m)+k, (j+1)*(m)+k+1);
            if(j+1<n && k-1>0 && a[j+k][k-1]==1)
                dus->union(j*(m)+k, (j+1)*m+k-1);
            if(j-1>=0 && k+1<m && a[j-1][k+1]==1)
                dus->union(j*(m)+k, (j-1)*m+k+1);
            if(j-1>=0 && k-1>=0 && a[j-1][k-1]==1)
                dus->union(j*(m)+k, (j-1)*m+k-1);
        }
    }
```


Pawit

```
int *C = new int [n*m];
int numberofIslands = 0;
~~for (int a~~
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) {
                numberofIsands ++;
                c[x]++;
            }
        else
            c[x]++;
    }
}
}
return numberofIslands;
}
```


Passed