Lab prog 3
ADS

Dhanraj . K
1BM18CS027

=> Finding the number of islands using disjoint sets

A :->

Step 1 -> set initial count_of_islands = 0

Step2 -> traverse all the indexes in 2D matrix

Step 3 -> If value at index is 1 thecheck all 8 neighbours, if a neighbour is also 1 take union of index & its neighbour

Step 4 -> define an array
=> arr [row * column]
and store frequences of all set

Step5 -> traverse the whole matrix again

Step6 : -> if a value is 1. find its set and check if frequency of set in above array is 0, if 0 then. increment, count_of_islands

```
//->

int count_islands( vector<vector<int>> a) {

        int n =  a.size();      //column size
        int m =  a[0].size();   // row size
        Disjoint set *d = new Disjointset(n*m);
        // step 3
         for ( int j=0; j<n; j++){
```

```
for (int k =0; k<n; k++){
        if (a[j][k]==0)  continue;
        // check all 8 neighbours and
        if do union if neighbour is also 1

    //Eg ->
    if (j+1 ~~<~~ n && a[j+1][k]==1)
            d-> union (j*(m)+k , (j+1)*(m)+k);
            ,
            :
    }
}
int no_of_islands =0;
int *arr = new int[n*m];
for ( int j=0 ;j<n; j++){
    for (int k=0 ; k<m ; k++) {
        if (a[j][k] ==1) {
            ~~if~~ int pos = dus->find (j*m+k);
            if(arr[x] ==0){
                no_of_islands ++;
            arr[pos] ++;
            }
            else
            arr[pos]++
        }
    }
  }
return no_of_islands;
}
```