NAME : NIHA

USN : 1BM18CS060

papergrid

Date: / /

07-Oct

LAB-3

No. of Islands using disjoint sets:-

Algorithm:-

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

2. Count all islands [illegible]

3. Loop through the matrix (2D) (mat[][])
If find an island x (points to root parent element s), check the adjacent neighbours. If any adjacent island present, it should be in the same subset of x.
If there is an adjacent island y and is not in the same subset of x, i.e., the root parent element of y is not s, then merge y to subset s by setting y as the parent element of s and count --
(Union operation)

4. While one island is merged to a subset, the number of island will be decremented by 1. After we unite all the connected islands, we get the no. of islands.

[illegible]

N[illegible]HA
1BM18CS060

papergrid
Date: / /

```
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+1][k-1] ==1)
                dus -> Union(j*m +k , (j+1)*(m) + k-1);
```

[illegible]

NIHA
1BM18CS060

papergrid
Date: / /

```
    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);
      }
}
    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;
}
```

[illegible]