ADA LAB-4

K. ARUN TEJA
1BM18CS041
5-A

```
1)  int count = 0

                                        ~~i = arr~~
~~2)  for i in arr:~~
          ~~for j in arr[0]:~~

2)    for each element in 2D matrix:
                if a[i][j] == 0
                        continue

            else
                check all the neighbours when a[i][j] ==1
                if neighbours of a[i][j] ==1 then
                                         present
                union ( neigbours , index )

3)   int freqnency [ ] = new int [n*m]      // n= columns  of 2D array
                                               m= rows

4)       for( int j=0; j<n; j++)
              for( ink k=0; k<n; k++)
                  if (a[j][k] ==1)
                        int x = Unionfind function (j*m+k);
                        if (c[x]==0)
                                count++;
                                c[x]++;

                        else
                            c[x]++

5) return count;

=>   union function (int a , int b) {

            int arep = find (a);
            int brep= find (b);

            if ( arep == brep)
                 return;
```

```
if ( rank [arep] < rank [brep])
                parent [ arep] = brep;

else if ( rank [ brep] < rank [arep])
                parent [brep] = arep;

    else
        parent [brep] = arep;
        rank [arep] = rank [arep] + 1;

=> find (int a)      function:
        if ( parent [a] != a)
                    return find (parent [a]);

        return a;
```