Chandan C Bagan
1BM18CS026

ADS - Lab4

```
class Islands{
                 int parent [100]; int count ;
 int find parent ( int x, int parent[ ])
 {
        if (parent [x] = = x )
            return x;

        return parent [x] = find ( parent [x]) ;
 }

  Void makeunion ( int x, int y )
  {
       int xroot = findparent (x) ;
       int yroot = find parent (y);
       if (xroot != yroot )
          parent [xroot ] = yroot ;
          count- - ;
        }
  }
  public void setcount (int n){
       count = n ;
   }
  public int count ( ) {
       return count ;
   };
};
  Int no of islands ( vector<vector<int>> matrix) {
    int count =0;
   int m = matrix.size ( ) ;
   int n = matrix [0] . size ( ) ;

   for ( int i= 0; i< m ; i++ ) {
        for ( int j=0 ; j<n ; j++ ) {
            if ( mat [i][j ])
                count ++ ;
   }  }
```

Chandan C Dog
1BM18CS026

Islands

```
~~unionfind~~ uf= new unionfind (m*n);
uf.setCount(count);

for(int i=0; i<m; i++){
    for(int j=0; j<n; j++){
      if (mat[i][j]){
        if (i>0 && matrix[i-1][j]){
          uf.makeunion (n*j+i, n*(i-1)+j);
      }
     if( i<m-1 && matrix[i+1][i]){
          uf.makeunion (n*i+j, n*(i+1)+j );
     }
    if (j>0 && matrix[i][j-1]){
          uf.makeunion (n*i+j, n*(i+1)+j);
    if (j<n-1 && matrix [i][j+1]){
          uf.makeunion( n*i+j , n*i+j+1);
     }
   if (i>0 && j>0 && matrix [i-1][j-1]){
          uf.~~connect~~ (n*i+j, n*(i-1)+j-1);
             makeunion
    }
   if (i<m-1 && j <matrix [i+1][j+1]){
        uf.makeunion (n*i+j, n*(i+1)+j+1);

   if (i>0 && j<n-1 && matrix [i-1][j+1]){
        uf.makeunion (n*i+j, n*(i-1)+ j+1);

  if (i<m-1 && j>0 && mat [i+1][j-1]){
        uf.~~connect~~ (n*i+j, n*(i+1)+j-1);
           makeunion
   }
 }
}
 return uf.count();
}
```