PRANAV JAGADEESH 1BM18CS071

PROGRAM - 3 WRITE - UP 5 B BATCH - 4

```
Class DisjointUnionSets
{
 vector <int> rank, parent
 int n
 public : DisjointUnionSets (int n)
      {
        rank.resize(n)
        parent.resize(n)
        this -> n <- n
        makeSet()
   }
   void makeSet()
  {
     for (int i <- 0, i<n, i++)
             parent[i] <- i
  }
  int find (int x)
  {
     if (parent[x] != x)
    {
      return find (parent[x])
    }
     return x
 }
  void Union (int x, int y)
  {
     int xRoot = find(x)
     int yRoot = find(y)
      if (xRoot ⇔ yRoot)
              return
       if (rank[xRoot] < rank[yRoot])
              parent[xRoot] ⇔ yRoot
```

```
else if(rank[yRoot]< rank[xRoot])
    parent[yRoot]<=> XRoot ;
else
{
    parent[yRoot]<=>xRoot
    rank[xRoot]<=> rank[xRoot]+1
}

}

}
int countIslands(vector<vector <int >>a)
{
    int n= a.size()
    int m =a[0].size()
    DisjointUnion Sets *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)
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
}

int main (void)
{  vector<vector<int> > a ⟷ {{1,1,0,0,0}, {1,0,1,0,1}
      {1,0,0,1,1}, {0,1,0,0,1}, {0,0,0,0,0}}
  cout << "Number of Islands is: " << countIslands (a) << endl
 }
```