Updated writeup (disjoint sets)

Arjun. A.S
1BM18CS019

```
class Islands
{ public:
      int parent[100];
      int count;

      Islands (int n)
    {
          parent[n];
          for (i=0; i<n; i++)
              parent[i]=i;
          count=0;
    }

    int findparent (int x)
    {
          if (parent[x] == x)
              return x;
        else
          return findparent(parent[x]);
    }
    void unions( int x, int y)
  {  int root_x = findparent(x);
     int root_y = findparent(y);
     if (root_x != root_y)
      {      parent[root_x] = root_y;
             count--;
      }
};

int noofislands (vector<vector<int>> grid)
{       int count=0;
        int a= grid.size();
        int b = grid[0].size();
```

```
for ( int i=0 ; i<a; i++)
  {    for (int j=0; j<b; j++)
        {           if ( guid [i] [j] != 0)
                       count ++ ;

        }
  }

Islands   iland= new Islands (a*b);
   iland.count  = count;

for (int  i=0; i<a ;i++)
   {  for (int  j=0 ;  j<b; j++)
       {
                if (grid [i][j] !=0)
           {
                 if (i>0 && grid [i-1][j] !=0)
                    iland. unions(b*i+j, (b*(i-1)+j))

                 if ( i<a-1 & & grid [i+1] (j] !=0)
                    iland. unions (b*i+j , b* (i+1)+j)
                 if (j>0 && gerid [i] [j-1] !=0 )
                    iland . unions (b*i+j , b* i+(j-1))

              if (j<b-1 && grid[i] [j+1] !=0)
                    iland . unions (b*i+j , b* i +j+1);

              if (i>0 && j>0  && grid [i-1][j-1] !=0)
                    iland . unions (b* i+j , b* (i-1)+j-1)
              if (i>0 && j<b-1 && grid [i-1] (j+1] !=0)
                    iland. unions (b*i+j, b*(i-1)+(j+1)
              if (i<a-1 && j<b-1  && grid [i+1] [j+1] !=0]
                    iland. unions (b* i+j, b* (i+1)* (j+1))
        }  }
   }
   return ;iland.count ;
}
```