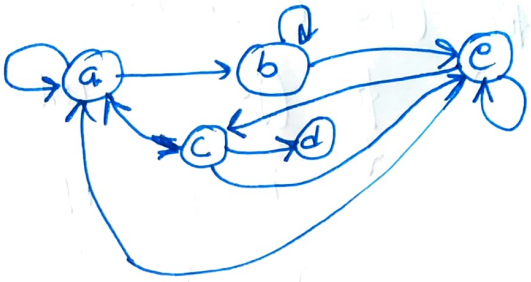
07/10/2020

ADS
Week 3 - Writeup

Arjun A.S
1BM18CS019

Q. Find no. of connected islands in given 2D boolean matrix

	a	b	c	d	e
a	1	1	0	0	0
b	0	1	0	0	1
c	1	0	0	1	1
d	0	0	0	0	0
e	1	0	1	0	1



Two sets are said to be disjoint if they don't have any element in common.

```
// pseudocode.
int findparent (int parent[], int x)
{       //finding parent of x in parent array.
        if parent[x] = x
                return x;   //root is found

        else
                return parent(x) = findparent (parent, parent[x])  //recursive call
}.
void unionsets (int x, int y, int parent[])
{       // set count = n.
        //finding the representatives (roots) for x & y by
        calling findparent function
                root_x = findparent (parent, x)
        // if elements are in same set, no need to unite
        if (root_x = root_y)
                return

        parent [root_x] = root_y

        //count variable stores the total no. of 1's in grid.
        count --;
}
```

```
int islandcount (int grid [][20], int x, int y)
{   int *C=new int [row * col];
    count = 0
    for (i ← 0 To row)
    {
        for (j ← 0 To col)
        {
            if (grid[i][j] == 1)
            {   // checking all eight neighbours.
                if (i+1<rows && grid [i+1][j] == 1)
                    unionsets ( j * (col)+i , (i+1) * (col) +j )

                // check all the 7 possibilities for the
                   current position which is an island.
                if (i-1 >=0 && a[i-1][j] == 1)
                    unionsets ( i * (row)+j , (i-1) * (row) + j)

                if (j+1 < col && a[i][j+1] == 1 )
                    unionsets (i * col + j , i * col + j+1)

                if (j-1 >=0 && j-1 >=0 && a[i-1][j-1] == 1)
                    unionsets (i * col + j , (i-1)* col + j-1)

                if (j-1) >=0 && a[i][j-1] == 1 )
                    unionsets ( i * (col) + j ; (i * (row) + j-1)
            }
        }
    } return count ;
```