

Karthik Venugopal

(BM18CS043)

classmate

DATE

PAGE

07/08/2020

## Number of Islands Using Disjoint set

Pseudocode

countIslands (int a[][])

{

    n = a.size();

    m = a[0].size();

    disjoint \*dsu = new disjoint(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)

                dsu → Union(j \* m + k,  
                                (j+1) \* m + k);

            if (j-1 >= 0 && a[j-1][k] == 1)

                dsu → Union(j \* m + k,  
                                (j-1) \* m + k);

            if (j < n-1 && a[j][k+1] == 1)

                if (k+1 < m && a[j][k+1] == 1)

                    dsu → Union(j \* m + k,  
                                j \* m + k+1);

            if (k-1 >= 0 && a[j][k-1] == 1)

                dsu → Union(j \* m + k,  
                                j \* m + k-1);

if ( $j+1 < n \wedge k+1 < m \wedge a[j+1][k+1] == 1$ )

$\text{dis} \rightarrow \text{Union}(j * m + k, (j+1) * m + k+1)$

if ( $j+1 < n \wedge k-1 \geq 0 \wedge a[j+1][k-1] == 1$ )

$\text{dis} \rightarrow \text{Union}(j * m + k, (j+1) * m + k-1)$

if ( $j-1 \geq 0 \wedge k+1 < m \wedge a[j-1][k+1] == 1$ )

$\text{dis} \rightarrow \text{Union}(j * m + k, (j-1) * m + k+1)$

if ( $j-1 \geq 0 \wedge k-1 \geq 0 \wedge a[j-1][k-1] == 1$ )

$\text{dis} \rightarrow \text{Union}(j * m + k, (j-1) * m + k-1)$

    }

int\* c = new int[n\*m];

int num = 0;

for (int i = 0; i < n; i++)

{

    for (int k = 0; k < m; k++)

    {

        if ( $a[j][k] == 1$ )

        {

            int x = dis → find( $j * m + k$ );

            if ( $c[x] == 0$ )

            {

                // number of nodes added

$c[x]++$ ;

            }

```
    cin  
    c[x]++;  
    }  
    }  
    }  
    } returns num  
    }
```

sum is an object of a class

~~sum~~ is a class which has a find function  
~~sum~~ and a union function. ~~which~~