

# Count Islands using disjoint sets

Krishna prasad V

CSM19CS017

7.10.20

count Islands (int arr[][2])

{  
    n = length(a) // rows and columns  
    m = length(a[0])

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

{ for (int k=0; k<m; k++)

{ if (a[j][k] == 0)  
    continue

// check 8 neighbours with union

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

set → union(j \* m + k, (j+1) \* m + k)

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

set → union(j \* m + k, (j-1) \* m + k)

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

set → union(j \* m + k, (j) \* m + k+1)

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

set → union(j \* m + k, (j) \* m + k-1)

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

set → union(j \* m + k, (j+1) \* m + k+1)

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

set → union(j \* m + k, (j+1) \* m + k-1)

if  $(j-1 > 0 \text{ \& \& } k-1 < m \text{ \& \& } a[j-1][k-1] == 1)$   
 set  $\rightarrow$  union (jump to, get to next)

if  $(j-1 > 0 \text{ \& \& } k-1 > 0 \text{ \& \& } a[j-1][k-1] == 1)$   
 set  $\rightarrow$  union (jump to, (j-1) < m+1)

}

}

// Counting any  
~~islands~~  $c[n+m]$

num of Islands = 0

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

{ for (int k=0; k<m; k++)

{ if  $(a[j][k] == 1)$

{ int x: set to (jump to)

if  $(c[x] == 0)$

{ num of Islands ++;

$c[x]++$ ;

}

else

$c[x]++$ ;

}

}

}

return num of Islands;

}