

~~int n,~~  
1BM18CS112

Sumanth-KV.

Batch-6

int n, m;

bool vis[1001][1001];

int arr[1001][1001];

int parent[1001][1001];

int par[1001];

bool isValid(int x, int y) {

if (x <= 0 || y <= 0 || x > n || y > m)

return false;

if (vis[x][y] == 1 || arr[x][y] == 0)

return false;

return true;

}

int find(int x) {

while (par[x] != x)

x = par[x];

return x;

}

void unionBoth(int x, int y, int part)

vis[x][y] = 1;

parent[x][y] = part;

int xr = find(x);

int yr = find(y);

if (xr < yr)

par[xr] = yr;

else par[yr] = xr;

```
if (isValid(x-1, y))
    union_both(x-1, y, part);
```

```
if (isValid(x-1, y+1))
    union_both(x-1, y+1, part);
```

```
if (isValid(x, y+1))
    union_both(x, y+1, part);
```

```
if (isValid(x+1, y+1))
    union_both(x+1, y+1, part);
```

```
if (isValid(x+1, y))
    union_both(x+1, y, part);
```

```
if (isValid(x+1, y-1)) y=1
    union_both(x+1, y-1, part);
```

```
if (isValid(x, y-1)) y=1
    union_both(x, y-1, part);
```

```
if (isValid(x-1, y-1))
    union_both(x-1, y-1, part);
```

```
}
```

```
main() {
    cin >> n >> m;
    for i=1 to n
        for j=1 to m
            cin >> arr[i][j]
            par[i] = i;
}
```

```

count = 0
for i = 1 to n
  for j = 1 to m
    if (vis[i][j] == 0 && arr[i][j] == 1)
      vis[i][j] = 1
      union_both(i, j, count)
    } } } count++
  } } }
for i = 1 to n
  for j = 1 to m
    if (parent[i][j] != -1)
      mp[parent[i][j]]++
    } } }
return mp.size();

```

③ Summation