**SOLUTION**

class Solution {

public:

int dfs(vector<vector<char>>& grid,int i,int j){

if(i<0 || i>=grid.size() || j<0 || j>=grid[i].size() || grid[i][j]=='0'){

return 0;

}

grid[i][j]='0';

dfs(grid,i+1,j);

dfs(grid,i-1,j);

dfs(grid,i,j+1);

dfs(grid,i,j-1);

return 1;

}

int numIslands(vector<vector<char>>& grid) {

if(grid.size()==0){

return 0;

}

int n=0;

for(int i=0;i<grid.size();i++){

for(int j=0;j<grid[i].size();j++){

if(grid[i][j]=='1'){

n+=dfs(grid,i,j);

}}}

return n; }};

**TIME COMPLEXITY : O(N\*M) if N==M then, O(N^2)**

**SPACE COMPLEXITY : O(1)**