Question: https://leetcode.com/problems/surrounded-regions/

I hope the comments wil help you get through.

Code:

class Solution {

public void solve(char[][] board) {

// Base condition

if(board.length == 0) return;

// 1st Loop : Traversing over top column & bottom column, to find any 'O' present by the boundary

for(int i = 0; i < board[0].length; i++){

if(board[0][i] == 'O'){

DFS(board, 0, i);

}

if(board[board.length - 1][i] == 'O'){

DFS(board, board.length - 1, i);

}

}

// 2nd Loop : Traversing over left row & right row, to find any 'O' present by the boundary

for(int i = 0; i < board.length; i++){

if(board[i][0] == 'O'){

DFS(board, i, 0);

}

if(board[i][board[0].length - 1] == 'O'){

DFS(board, i, board[0].length - 1);

}

}

// 3rd Loop : Now in this we will traverse on each n every node & check if they are 'O' convert into 'X', if they are '@' convert into 'O'

for(int i = 0; i < board.length; i++){

for(int j = 0; j < board[0].length; j++){

if(board[i][j] == 'O'){

board[i][j] = 'X';

}

else if(board[i][j] == '@'){

board[i][j] = 'O';

}

}

}

return;

}

// This calls helps in convert the 'O' node present near by the boundary convert them into '@'

public void DFS(char[][] board, int i, int j){

if(i < 0 || i >= board.length || j < 0 || j >= board[0].length || board[i][j] != 'O'){

return;

}

board[i][j] = '@';

DFS(board, i + 1, j);

DFS(board, i - 1, j);

DFS(board, i, j + 1);

DFS(board, i, j - 1);

}

}

Github Link :<https://lnkd.in/ecwtJeaz>