



AfterAcademy



OFFER

Android Online Course by MindOrks

Start your career in Android Development.
Learn by doing real projects.

[ENROLL NOW](#)


Admin AfterAcademy
22 Sep 2020

Surrounded regions

Interview question

Surrounded regions

X X O O X O
O X O X X X

Asked in 

afteracademy.com

This website uses **cookies** to ensure you
get the best experience on our website.

[READ MORE](#)[GOT IT!](#)

Difficulty: Hard

Asked in: Google

Understanding The Problem

Problem Description

Given a 2-D matrix `board` where every element is either 'O' or 'X', write a program to replace 'O' with 'X' if surrounded by 'X'.

Problem Note

- An 'O' or a set of 'O' is considered to be surrounded by 'X' if there are 'X' at locations just below, just above, just left and just right of it.
- Surrounded regions shouldn't be on the border, which means that any 'O' on the border of the board are not flipped to 'X'.
- Any 'O' that is not on the border and it is not connected to an 'O' on the border will be flipped to 'X'.
- Two cells are connected if they are adjacent cells connected horizontally or vertically.

Example 1

```
Input: board[][] = [
    ['X', 'O', 'X', 'X', 'X', 'X'],
    ['X', 'O', 'X', 'X', 'O', 'X'],
    ['X', 'X', 'X', 'O', 'O', 'X'],
    ['O', 'X', 'X', 'X', 'X', 'X'],
    ['X', 'X', 'X', 'O', 'X', 'O'],
    ['O', 'O', 'X', 'O', 'O', 'O'],
]
```

```
Output: board[][] = [
    ['X', 'O', 'X', 'X', 'X', 'X'],
    ['X', 'O', 'X', 'X', 'X', 'X'],
    ['X', 'X', 'X', 'X', 'X', 'X'],
    ['X', 'X', 'X', 'X', 'X', 'X'],
    ['X', 'X', 'X', 'X', 'X', 'X'],
    ['X', 'X', 'X', 'X', 'X', 'X'],
]
```

This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)

Example 2

```
Input: board[][] = [
  ['X', 'X', 'X', 'X']
  ['X', 'O', 'X', 'X']
  ['X', 'O', 'O', 'X']
  ['X', 'O', 'X', 'X']
  ['X', 'X', 'O', 'O']
]
Output: board[][] = [
  ['X', 'X', 'X', 'X']
  ['X', 'X', 'X', 'X']
  ['X', 'X', 'X', 'X']
  ['X', 'X', 'X', 'X']
  ['X', 'X', 'O', 'O']
]
```

Solutions

This problem can be solved by the famous **Flood Fill Algorithm**. The main difference here is that a 'O' is not replaced by 'X' if it lies in a region that ends on a boundary.

Flood fill algorithm can be simply explained as a graph traversal problem, representing the given area as a matrix and considering every cell of that matrix as a vertex that is connected to points above it, below it, to the right of it, and to the left of it. In general, any cell $[x,y]$ connected with $[x-1][y]$, $[x+1][y]$, $[x][y-1]$, $[x][y+1]$ cells.

For this problem, the
will convert all the m

This website uses **cookies** to ensure you
get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)

rsive function that

```

void floodFillUtil(char[][] board, int x, int y, char markA, char markB, int dimsX, int dimsY)
{
    if (x < 0 or x >= dimsX or y < 0 or y >= dimsY)
        return
    if (board[x][y] != markA)
        return
    // Replace the mark
    board[x][y] = markB
    // Recur for the four directions
    floodFillUtil(board, x+1, y, markA, markB, dimsX, dimsY)
    floodFillUtil(board, x-1, y, markA, markB, dimsX, dimsY)
    floodFillUtil(board, x, y+1, markA, markB, dimsX, dimsY)
    floodFillUtil(board, x, y-1, markA, markB, dimsX, dimsY)
}

```

So, The idea is very simple, we will convert all the 'O' to '\$' . Now we will search for all the '\$' at the boundary of the matrix and for each such '\$' we will apply the flood fill algorithm to convert them to 'O' . In this way, we will be left with all the '\$' which will be surrounded by only 'X' .

You can try the [problem here](#).

Example 1 can be visualized as

```

['X', 'O', 'X', 'X', 'X', 'X']
['X', 'O', 'X', 'X', 'O', 'X']
['X', 'X', 'X', 'O', 'O', 'X']
['O', 'X', 'X', 'X', 'X', 'X']
['X', 'X', 'X', 'O', 'X', 'O']
['O', 'O', 'X', 'O', 'O', 'O']

```

convert all 0 to \$

```

['X', '$', 'X', 'X', 'X', 'X']
['X', '$', 'X', 'X', '$', 'X']
['X', 'X', 'X', '$', '$', 'X']
['$', 'X', 'X', 'X', 'X', 'X']
['X', 'X', 'X', '$', 'X', '$']
['$', '$', 'X', '$', '$', '$']

```

Choose boundry \$ and

```
['X', 'O', 'X', 'X',
```

This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)

```
['X', 'O', 'X', 'X', '$', 'X']
['X', 'X', 'X', '$', '$', 'X']
['O', 'X', 'X', 'X', 'X', 'X']
['X', 'X', 'X', 'O', 'X', 'O']
['O', 'O', 'X', 'O', 'O', 'O']
```

Replace all the \$ with X

```
['X', 'O', 'X', 'X', 'X', 'X']
['X', 'O', 'X', 'X', 'X', 'X']
['X', 'X', 'X', '$', 'X', 'X']
['O', 'X', 'X', 'X', 'X', 'X']
['X', 'X', 'X', 'O', 'X', 'O']
['O', 'O', 'X', 'O', 'O', 'O']
```

Solution steps

- Create a flood fill function, which recursively marks each node with the input symbol in DFS fashion until it touches the boundary where the boundary will be the end of the matrix or boundary of different symbols.
- Convert each 'O' to '\$'
- Apply flood fill for all the boundary '\$' to convert into 'O'
- Now replace all the left '\$' into 'X'

Pseudo Code

```
void floodFillUtil(char[][] board, int x, int y, char markA, char markB)
{
    if (x < 0 or x >= board[0].size() or y < 0 or y >= board.size)
        return
    if (board[x][y] != markA)
        return
    // Replace the mark
    board[x][y] = markB
    // Recur for the four directions
    floodFillUtil(board, x+1, y, markA, markB)
    floodFillUtil(board, x-1, y, markA, markB)
    floodFillUtil(board, x, y+1, markA, markB)
    floodFillUtil(board, x, y-1, markA, markB)
}
```

This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)

```

void surroundedRegions(char[][] board, int dimsX, dimsY)
{
    for (int i=0 to i < dimsX)
        for (int j=0 to j < dimsY)
            if (board[i][j] is 'O')
                board[i][j] = '$'
    for (int i=0 to i < dimsX) // Left boundary
        if (board[i][0] is '$')
            floodFillUtil(board, i, 0, '$', 'O')
    for (int i=0 to i < dimsX) // Right boundary
        if (board[i][dimsY - 1] is '$')
            floodFillUtil(board, i, dimsY - 1, '$', 'O')
    for (int i=0 to i < dimsY) // Top boundary
        if (board[0][i] is '$')
            floodFillUtil(board, 0, i, '$', 'O')
    for (int i=0; i < dimsY; i++) // Bottom boundary
        if (board[dimsX - 1][i] is '$')
            floodFillUtil(board, dimsX - 1, i, '$', 'O')
    for (int i=0 to i < dimsX)
        for (int j=0 to j < dimsY)
            if (board[i][j] is '$')
                board[i][j] = 'X'
}

```

Complexity Analysis

Time Complexity: $O(n^2)$

Space Complexity: $O(n^2)$

where n^2 is the size of the board matrix.

Critical Ideas To Think

- What does the flood fill algorithm do?
- How come we assumed the matrix as a graph and considered adjacent nodes as connected nodes?
- Why did we first
- Floodfill acts like

This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)

o you think,

converting it into a BFS algorithm will affect the solution?



NEW

Android App Development Online Course by MindOrks

Start your career in Android Development. Learn by doing real projects.

[CHECK NOW](#)

Suggested Problems To Solve

- Number of islands
- Flood fill an image
- Determine the perimeter of the island if 1's in the matrix represent an island and 0's represent water.
- Implement a fill function for the paint's color fill tool.

Please comment down below if you have a better insight in the above approach.

Happy Coding, Enjoy Algorithms!

This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)



NEW

Google Android Developer Interview

Google Android Engineer Interview Process and Preparation.

[CHECK NOW](#)

Share this blog and spread the knowledge



SHARE ON FACEBOOK



SHARE ON TWITTER



SHARE ON LINKEDIN



SHARE ON TELEGRAM



SHARE ON REDDIT



SHARE ON WHATSAPP

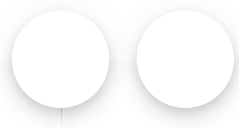
Our Learners Work At



This website uses **cookies** to ensure you get the best experience on our website.

[READ MORE](#)[GOT IT!](#)

Stay up to date. Follow us on



© Copyright 2019

MindOrks Nextgen Private Limited
Gurgaon, Haryana, India
+91-8287460223

About Us

MindOrks
Amit Shekhar
Janishar Ali

Quick Links

[Contact Us](#)
[Privacy Policy](#)
[Terms And Conditions](#)
[Cookie Policy](#)

Free Resources

[Publication](#)
[Medium](#)
[Video Lessons](#)
[Open Source](#)

This website uses **cookies** to ensure you
get the best experience on our website.

[READ MORE](#)

[GOT IT!](#)