

Problem Statement:

Given an $m \times n$ matrix board containing 'X' and 'O', Capture all regions that are 4-directionally surrounded by 'X'.

A region is captured by flipping all 'O's to 'X's in that surrounded region

Sample input:

board = [["X","X","X","X"], ["X","O","O","X"], ["X","X","O","X"], ["X","O","X","X"]]

Output: [["X","X","X","X"], ["X","X","X","X"], ["X","X","X","X"], ["X","O","X","X"]]

My Approach:

There are different ways to solve this problem . But I used the Reverse approach.

Since the boundary O's are non surrounded,I ran the DFS algorithm and converted the them to some character for say '#'.

Output for this case :

[["X","X","X","X"], ["X","O","O","X"], ["X","X","O","X"], ["X","X","#", "X"]]

Now all the remaining Os must be surrounded by X 4- directionally. So I changed them to X by simply using For loop.

Output for this case :

[["X","X","X","X"], ["X","X","X","X"], ["X","X","X","X"], ["X","X","#", "X"]]

In last i converted the '#' back to O using For and the problem is solved

Final Output :

```
[["X","X","X","X"],["X","X","X","X"],["X","X","X","X"],["X","O","X","X"]]
```