Surrounded Regions

Question: Given a 2D board containing 'X' and 'O', capture all regions surrounded by 'X'.

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

For example:

X X X X

X O O X

X X O X

X O X X

After running your function, the board should be:

X X X X

X X X X

X X X X

X O X X

Solutions:

```
import collections
class Solution:
    def solve(self, board):
        if board == []: return []
        lineNum = len(board)
        colNum = len(board[0])
        queue = collections.deque()
        visited = [[False for j in range(colNum)] for i in range(lineNum)]
        for i in range(colNum):
            if board[0][i] == 'O': queue.append((0, i))
            if board[lineNum-1][i] == 'O': queue.append((lineNum - 1, i))
```

```
        for i in range(1, lineNum - 1):
            if board[i][0] == 'O': queue.append((i, 0))
            if board[i][colNum-1] == 'O': queue.append((i, colNum - 1))
        while queue:
            t = queue.popleft()
            if board[t[0]][t[1]] == 'O': board[t[0]][t[1]] = '$'
            visited[t[0]][t[1]] = True
            if t[0] + 1 < lineNum and board[t[0] + 1][t[1]] == 'O' and visited[t[0] +
1][t[1]] == False:
                queue.append((t[0] + 1, t[1]))
            if t[0] - 1 >= 0 and board[t[0] - 1][t[1]] == 'O' and visited[t[0] - 1][t[1]] ==
False:
                queue.append((t[0] - 1, t[1]))
            if t[1] + 1 < colNum and board[t[0]][t[1] + 1] == 'O' and visited[t[0]][t[1] +
1] == False:
                queue.append((t[0], t[1] + 1))
            if t[1] - 1 >= 0 and board[t[0]][t[1] - 1] == 'O' and visited[t[0]][t[1] - 1] ==
False:
                queue.append((t[0], t[1] - 1))
        for i in range(lineNum):
            for j in range(colNum):
                if board[i][j] == 'O': board[i][j] = 'X'
                if board[i][j] == '$': board[i][j] = 'O'
        return board


Solution().solve([["X","X","X","X"],
                  ["X","O","O","X"],
                  ["X","X","O","X"],
                  ["X","O","X","X"]])
```