Approach:

1. Find all border cells containing “O”
2. Perform bfs on these cells so that all groups of “O” coinciding with border cells(not surrounded with “x”) is omitted by marking them as “T”
3. Rest of the cells containing “O” is marked as “x”
4. Reset “T” back to “O”

Time -> O(mn)

Code:

r = len(board)

c= len(board[0])

def dfs(u,v):

if (u<0 or u>=r) or (v<0 or v>=c):

return

if board[u][v]!="O":

return

# mark current cell as T

board[u][v] = 'T'

# take all neighbours of u,v

nei = [ (u-1,v), (u,v-1), (u,v+1), (u+1,v)]

# call dfs on each neighbour

for x,y in nei:

dfs(x,y)

# BASE:

if not board:

return 0

# top row and bottom row

for i in range(c):

dfs(0,i)

dfs(r-1, i)

# left column and ight col

for j in range(r):

dfs(j, 0)

dfs(j, c-1)

# mark remaining O's with X

# Remark T's back to O's

for i in range(r):

for j in range(c):

if board[i][j] == "T":

board[i][j] = "O"

else:

board[i][j] = "X"