

Sisihir's Approach

Take a n and m(matrix dimensions) input from the user.

Create a matrix of n*m size. This matrix contains 'X' and 'O'.

Now according to the question, the need is to flip all O's that are 4 directionally surrounded by 'X'.

call DFS from each cell on the boundary of the board which contains 'O'.

While performing Dfs, mark all the cells(O's) as -1. -1 indicates that these cells will not be converted to 'X' since they are connected to the boundary.

Now the O's that are left are the target. Convert all these leftover O's to 'X'.

Now traverse the matrix again and convert all '-1s' back to 'O'. These O's will not be converted, so changing them back.

Shikhar's Approach

In this problem we need to understand what exactly surrounded by 'X' means. It actually means that if we start from 'O' at the border, and we traverse only 'O', only those 'O' are not surrounded by 'X'. So the plan is the following:

Start dfs or bfs from all 'O', which are on the border. When we traverse them, let us color them as '-1', a temporary color. Now, when we traverse all we want, all colors which are not '-1' need to be renamed to 'X' and all colors which are '-1' need to be renamed to 'O', and that is all! Complexity: time complexity is $O(m \times n)$, where m and n are sizes of our board. Additional space complexity can also go upto $O(m \times n)$ to keep stack of recursion.