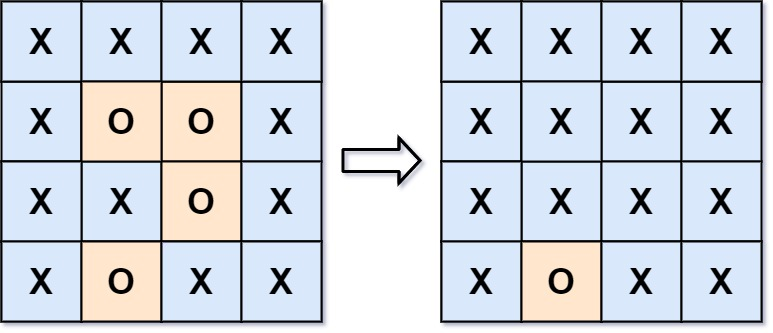
**Documentation for python assignment**

**Problem Statement: Surrounded Region**

Given an m\*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 into 'X's in that surrounded region.

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"]]

**Approach to Problem:**

**Step 1:** Take matrix as input.

**Step 2:** Used recursive call at boundary of matrix for.

**Step 3:** In recursive call ,if index of matrix is not in range of matrix index or if it is not ‘O’ then return your recursive call. During the recursive call replace ‘O’ with ‘P’ .This prevents an infinite recursive call.

**Step 4:** In the recursive call ,I have used four directions to traverse from the current position and did a recursive call for each of them.

**Step 5:** The left out ‘0’ in the middle of the matrix is anyway surrounded by ‘X’. So replace them with ‘X’.

**Step 6:** Now at last the boundary connected ‘P’ is replaced with ‘O’.Because basically They are not surrounded by ‘x’.

**Time complexity:** 0(M\*N) ; where M is row length and N is column length.

**Space complexity:**0(M\*N); auxiliary space used for recursive calls.