```
class Solution {
    int[][] dirs = {
                    {0, 1}, // down
                    {0, -1}, // up
                    {1, 0}, // right
                    {-1, 0}, // left
                    {1, 1},
                    {1, -1},
                    {-1, 1},
                    {-1, -1}
            };

    public char[][] updateBoard(char[][] board, int[] click) {
        if (board[click[0]][click[1]] == 'M') {
            board[click[0]][click[1]] = 'X';
            return board;
        }
        dfs(board, click[0], click[1], board.length, board[0].length);
        return board;
    }

    private void dfs(char[][] board, int i, int j, int m, int n) {
        if (i < 0 || i >= m || j < 0 || j >= n || board[i][j] != 'E') {
            return;
        }
        int neighbor = neighMines(board, i, j, m, n);
        if (neighbor == 0) {
            board[i][j] = 'B';
            for (int[] dir : dirs) {
                dfs(board, i+dir[0], j+dir[1], m, n);
            }
        } else {
            board[i][j] = (char)(neighbor + '0');
        }
    }

    private int neighMines(char[][] board, int i, int j, int m, int n) {
                    int res = 0;

                    for (int x = Math.max(i - 1, 0); x < Math.min(i + 2, m); x++) {
                            for (int y = Math.max(j - 1, 0); y < Math.min(j + 2, n); y++) {
                                    if (board[x][y] == 'M') res++;
                            }
                    }

                    return res;
            }
}
```

You are given a 2D char matrix representing the game board. **'M'** represents an **unrevealed** mine, **'E'** represents an **unrevealed** empty square, **'B'** represents a **revealed** blank square that has no adjacent (above, below, left, right, and all 4 diagonals) mines, **digit** ('1' to '8') represents how many mines are adjacent to this **revealed** square, and finally **'X'** represents a **revealed** mine.

Now given the next click position (row and column indices) among all the **unrevealed** squares ('M' or 'E'), return the board after revealing this position according to the following rules:

1. If a mine ('M') is revealed, then the game is over - change it to **'X'**.
2. If an empty square ('E') with **no adjacent mines** is revealed, then change it to revealed blank ('B') and all of its adjacent **unrevealed** squares should be revealed recursively.
3. If an empty square ('E') with **at least one adjacent mine** is revealed, then change it to a digit ('1' to '8') representing the number of adjacent mines.
4. Return the board when no more squares will be revealed.