1020. Number of Enclaves

1020. Number of Enclaves

You are given an m x n binary matrix grid, where 0 represents a sea cell and 1 represents a land cell.

A move consists of walking from one land cell to another adjacent (4-directionally) land cell or walking off the boundary of the grid.

Return the number of land cells in grid for which we cannot walk off the boundary of the grid in any number of moves.

```
class Solution {
public:
    int numEnclaves(vector<vector<int>>& grid) {
        queue<pair<int, int>> q;
        int n = grid.size();
        int m = grid[0].size();
        int vis[n][m] ;
         // Initialize vis array
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                vis[i][j] = 0;
            }
        }
        for(int i = 0; i<n; i++){
            for(int j = 0; j<m; j++){
                // first row, first col, last row, last col
                if(i==0 || j==0 || i==n-1 || j==m-1){
                    if(grid[i][j] == 1){
                        q.push({i, j});
                        vis[i][j] = 1;
                    }
                }
            }
        }

        int delrow[] = {-1, 0, +1, 0};
        int delcol[] = {0, +1, +0, -1};

        while(!q.empty()){
            int row = q.front().first;
            int col = q.front().second;
            q.pop();

            //traverse all 4 directions
            for(int i =0; i<4; i++){
                int nrow = row + delrow[i];
                int ncol = col + delcol[i];
                if(nrow >=0 && nrow<n && ncol>=0 && ncol <m && vis[nrow][ncol] == 0 &
& grid[nrow][ncol] ==1){
                    q.push({nrow, ncol});
                    vis[nrow][ncol] = 1;
                }
            }
        }
        int cnt = 0;
        for(int i =0; i<n; i++){
            for(int j=0; j<m; j++){
                if(grid[i][j] == 1 && vis[i][j] ==0) cnt++;
            }
        }
        return cnt;
    }
};
```