

827. Making A Large Island

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

In a 2D grid of 0s and 1s, we change at most one 0 to a 1.

After, what is the size of the largest island? (An island is a 4-directionally connected group of 1s).

Example 1:

Input: [[1, 0], [0, 1]]

Output: 3

Explanation: Change one 0 to 1 and connect two 1s, then we get an island with area = 3.

Example 2:

Input: [[1, 1], [1, 0]]

Output: 4

Explanation: Change the 0 to 1 and make the island bigger, only one island with area = 4.

Example 3:

Input: [[1, 1], [1, 1]]

Output: 4

Explanation: Can't change any 0 to 1, only one island with area = 4.

Notes:

- $1 \leq \text{grid.length} = \text{grid}[0].\text{length} \leq 50$.
- $0 \leq \text{grid}[i][j] \leq 1$.

Seen this question in a real interview before?

- [Subscribe](#) to see which companies asked this question.

Related Topics DFS

```
int dfs(vector<vector<int>>& grid, vector<vector<int>>& visited,
int x, int y)
{
    int m = grid.size();
    int n = grid[0].size();
    if(x>=m || x<0 || y>=n || y<0) return 0;
    if(grid[x][y]==0) return 0;
```

```

    if(visited[x][y]) return 0;
    visited[x][y]=1;
    int ret = 1;
    ret += dfs(grid,visited,x+1,y);
    ret += dfs(grid,visited,x-1,y);
    ret += dfs(grid,visited,x,y+1);
    ret += dfs(grid,visited,x,y-1);
    return ret;
}

int largestIsland(vector<vector<int>>& grid) {
    int m = grid.size();
    int n = grid[0].size();
    bool hasZero = false;
    int maxval = -1;

    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(grid[i][j]==0)
            {
                grid[i][j] = 1;
                hasZero=true;
                vector<vector<int>> visited(m,vector<int>(n,0));
                int ans = dfs(grid,visited,i,j);
                if(ans>maxval) maxval = ans;
                if(maxval == m*n) return maxval;
                grid[i][j] = 0;
            }
        }
    }
    if(!hasZero) return m*n;
    return maxval;
}

```