

361. Bomb Enemy

Given a 2D grid, each cell is either a wall 'Y', an enemy 'X' or empty '0' (the number zero), return the maximum enemies you can kill using one bomb.

The bomb kills all the enemies in the same row and column from the planted point until it hits the wall since the wall is too strong to be destroyed.

Note that you can only put the bomb at an empty cell.

Example:

For the given grid

```
0 X 0 0
X 0 Y X
0 X 0 0
```

return 3. (Placing a bomb at (1,1) kills 3 enemies)

```
/* C++
 * Author:ZZW
 */
#include <iostream>
#include <vector>
#include <map>
#include <string>
#include <sstream>
using namespace std;

int DFS(vector<vector<char> > &grid,int x,int y,int dx,int dy)
{
    int m=grid.size();
    int n=grid[0].size();
    if(x<0||x>=m||y<0||y>=n) return 0;
    int val=DFS(grid,x+dx,y+dy,dx,dy);
    return grid[x][y]=='X'?val+1:val;
}

int maxKilledEnimies(vector<vector<char> > &grid)
{
    if(grid.size()==0) return 0;
    int m=grid.size();
    int n=grid[0].size();
    int i,j,cnt;
    map<int,int>col,row;
    int Max=0;
    for(i=0;i<n;i++)
    {
        cnt=0;
        for(j=0;j<m;j++)
        {
            if(grid[j][i]=='X') cnt++;
            else if(grid[j][i]=='Y')break;
        }
        if(j==m) col[i]=cnt;
    }
    for(i=0;i<m;i++)
    {
        cnt=0;
        for(j=0;j<n;j++)
```

```

        {
            if(grid[i][j]=='X') cnt++;
            else if(grid[i][j]=='Y') break;
        }
        if(j==n) row[i]=cnt;
        //search
        for(j=0;j<n;j++)
        {
            if(grid[i][j]!='0') continue;
            if(col.count(j)) cnt=col[j];
            else
            {
                cnt=DFS(grid,i-1,j,-1,0)+DFS(grid,i+1,j,1,0);
            }
            if(row.count(i)) cnt+=row[i];
            else
            {
                cnt+=DFS(grid,i,j-1,0,-1)+DFS(grid,i,j+1,0,1);
            }
            Max = max(Max,cnt);
        }
    }
    return Max;
}

int main(int argc,char *argv[])
{
    /*
    0 X 0 0
    X 0 Y X
    0 X 0 0
    */
    char c[][4] = {{'0','X','0','0'},
                   {'X','0','Y','X'},
                   {'0','X','0','0'}};
    vector<vector<char> > path(3,vector<char>(4,'0'));
    for(int k=0;k<3;k++)
    {
        for(int j=0;j<3;j++)
        {
            path[k][j] = c[k][j];
        }
    }
    int ans = maxKilledEnimies(path);
    cout << ans << endl;
    return 0;
}

```