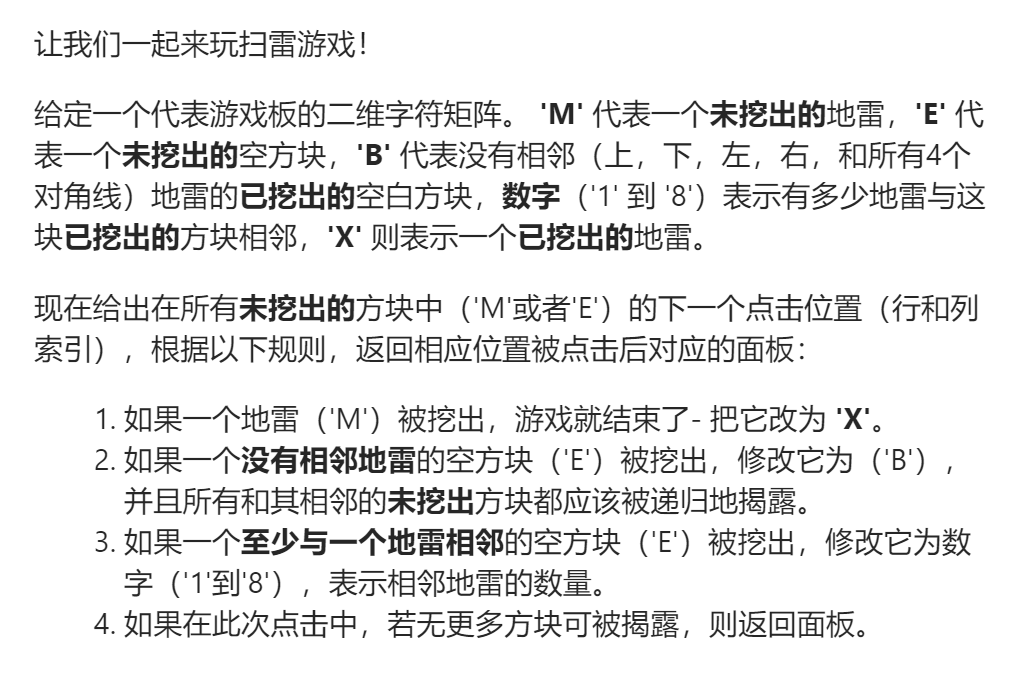
# [529. 扫雷游戏]

## 题目描述：



## 代码处理：

class Solution {  
public:  
 vector<vector<char>> updateBoard(vector<vector<char>>& board, vector<int>& click) {  
 //广度优先，用队列  
 pair<int,int>temp;  
 temp.first=click[0];  
 temp.second=click[1];  
 queue<pair<int,int>>que;  
 que.push(temp);  
 int m=board.size();  
 int n=board[0].size();  
  
 while(!que.empty()){  
 temp=que.front();  
 que.pop();  
 int x=temp.first,y=temp.second;  
 bool flag=false;  
 int nums=0;  
 if(board[x][y]=='E'){  
 board[x][y]='B';  
 if(x+1<m&&board[x+1][y]=='M'){  
 flag=true;  
 nums++;  
 }  
 if(x-1>=0&&board[x-1][y]=='M')flag=true,nums++;  
 if(y+1<n&&board[x][y+1]=='M')flag=true,nums++;  
 if(y-1>=0&&board[x][y-1]=='M')flag=true,nums++;  
 if(x+1<m&&y+1<n&&board[x+1][y+1]=='M')flag=true,nums++;  
 if(x+1<m&&y-1>=0&&board[x+1][y-1]=='M')flag=true,nums++;  
 if(x-1>=0&&y+1<n&&board[x-1][y+1]=='M')flag=true,nums++;  
 if(x-1>=0&&y-1>=0&&board[x-1][y-1]=='M')flag=true,nums++;  
  
 if(!flag){  
  
 if(x+1<m&&y+1<n&&board[x+1][y+1]=='E'){  
 pair<int,int>tp(x+1,y+1);  
 que.push(tp);  
 }  
 if(x+1<m&&y-1>=0&&board[x+1][y-1]=='E'){  
 pair<int,int>tp(x+1,y-1);  
 que.push(tp);  
 }  
 if(x-1>=0&&y+1<n&&board[x-1][y+1]=='E'){  
 pair<int,int>tp(x-1,y+1);  
 que.push(tp);  
 }  
 if(x-1>=0&&y-1>=0&&board[x-1][y-1]=='E'){  
 pair<int,int>tp(x-1,y-1);  
 que.push(tp);  
 }  
 if(x+1<m){  
 if(board[x+1][y]=='E'){  
 pair<int,int>tp(x+1,y);  
 que.push(tp);  
 }  
 }  
 if(x-1>=0){  
 if(board[x-1][y]=='E'){  
 pair<int,int>tp(x-1,y);  
 que.push(tp);  
 }  
 }  
 if(y+1<n){  
 if(board[x][y+1]=='E'){  
 pair<int,int>tp(x,y+1);  
 que.push(tp);  
 }  
 }  
 if(y-1>=0){  
 if(board[x][y-1]=='E'){  
 pair<int,int>tp(x,y-1);  
 que.push(tp);  
 }  
 }  
  
 }  
 }else if(board[x][y]=='M'){  
 board[x][y]='X';  
 break;  
 }  
  
 if(flag){  
 board[x][y]='0'+nums;  
 }  
 }  
 return board;  
 }  
};

## 代码优化：

class Solution {  
public:  
 int dir\_x[8] = {0, 1, 0, -1, 1, 1, -1, -1};  
 int dir\_y[8] = {1, 0, -1, 0, 1, -1, 1, -1};//此处很妙  
  
 void dfs(vector<vector<char>>& board, int x, int y) {  
 int cnt = 0;  
 for (int i = 0; i < 8; ++i) {  
 int tx = x + dir\_x[i];  
 int ty = y + dir\_y[i];  
 if (tx < 0 || tx >= board.size() || ty < 0 || ty >= board[0].size()) {  
 continue;  
 }  
 // 不用判断 M，因为如果有 M 的话游戏已经结束了  
 cnt += board[tx][ty] == 'M';  
 }  
 if (cnt > 0) {  
 // 规则 3  
 board[x][y] = cnt + '0';  
 } else {  
 // 规则 2  
 board[x][y] = 'B';  
 for (int i = 0; i < 8; ++i) {  
 int tx = x + dir\_x[i];  
 int ty = y + dir\_y[i];  
 // 这里不需要在存在 B 的时候继续扩展，因为 B 之前被点击的时候已经被扩展过了  
 if (tx < 0 || tx >= board.size() || ty < 0 || ty >= board[0].size() || board[tx][ty] != 'E') {  
 continue;  
 }  
 dfs(board, tx, ty);  
 }  
 }  
 }  
  
 vector<vector<char>> updateBoard(vector<vector<char>>& board, vector<int>& click) {  
 int x = click[0], y = click[1];  
 if (board[x][y] == 'M') {  
 // 规则 1  
 board[x][y] = 'X';  
 } else {  
 dfs(board, x, y);  
 }  
 return board;  
 }  
};  
  
作者：LeetCode-Solution  
链接：https://leetcode-cn.com/problems/minesweeper/solution/sao-lei-you-xi-by-leetcode-solution/  
来源：力扣（LeetCode）  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。