

## 289. Game of Life

---

题目描述: <https://leetcode.com/problems/game-of-life/>

给定一个二维空间（用数组表示），每个地方生活着一个细胞（活着1，死掉0）。细胞下个时刻存活状态根据周围8个邻居决定。

1. 如果本细胞是活着的，周围有少于两个细胞活着，则下一状态变成死细胞。
2. 如果本细胞是活着的，周围有多于三个细胞活着，则下一状态变成死细胞。
3. 如果本细胞是活着的，周围有两个活着三个细胞活着，则下一状态继续存活。
4. 如果本细胞是死的，周围有三个细胞活着，则下一状态变成活着的。

空间复杂度 $O(n)$ .

解题思路：

重点在于如何解决更新问题，因为就两种状态0，1，而且传入参数是整数二维数组，因此我们可以用两个bit，第一个bit表示下一状态，第二个表示当前状态。

墙壁问题，使用两层循环判断依次判断哪些细胞可以影响本细胞

代码：

```

class Solution {
public:
    void gameOfLife(vector<vector<int>>& board) {
        int m = board.size(), n = m > 0?board[0].size():0;
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                int count = 0 - board[i][j];
                for(int k = max(0, i-1); k <= min(m-1, i+1); k++){
                    for(int l = max(0, j-1); l <= min(n-1, j+1); l++){
                        count += board[k][l] & 1;
                    }
                }
                if(board[i][j]){
                    if(count == 2 || count == 3)
                        board[i][j] |= 2;
                }
                else if(count == 3){
                    board[i][j] |= 2;
                }
            }
        }
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                board[i][j] = board[i][j]>>1;
            }
        }
    }
};

```