

Game of Life

In-place

- 상태 변화를 비트로 표현하면 in-place로 간단히 풀 수 있습니다.
- 다음과 같이 상태를 정의합니다. 첫번째 비트는 현재 상태, 두번째 비트는 다음 상태가 되며, 숫자 0은 die, 1은 live를 표현합니다.

00(0): die -> die (count < 2 || count > 3)

01(1): live -> die (count < 2 || count > 3)

10(2): die -> live (count == 3)

11(3): live -> live (count == 2 || count == 3)

- 현재 셀의 두번째 비트(다음 상태)는 주위 8셀의 첫번째 비트(현재 상태)에만 영향을 받으며, 현재 셀의 첫번째 비트는 영향을 받지 않아 현재 상태를 유지할 수 있습니다.

Code

비트 연산을 통해 두번째 비트만 결과값으로 취함

```
void gameOfLife(vector<vector<int>>& board) {
    m = board.size();
    n = board[0].size();

    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            liveOrDie(board, i, j);
        }
    }

    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            board[i][j] = board[i][j] >> 1;
        }
    }
}
```

비트 상태 전이를 구현한 함수

```
void liveOrDie(vector<vector<int>>& board, int x, int y)
{
    int count = 0;
    for (int cx = x - 1; cx < x + 2; cx++)
    {
        for (int cy = y - 1; cy < y + 2; cy++)
        {
            if (cx == x && cy == y) continue;
            if (isInBoard(cx, cy))
            {
                if (board[cx][cy] & 1) count++;
            }
        }
    }
    if (board[x][y] & 1)
    {
        if (count < 2 || count > 3) board[x][y] = 1;
        else board[x][y] = 3;
    }
    else
    {
        if (count == 3) board[x][y] = 2;
    }
}
```

Link

- <https://github.com/tolerance93/LeetCode/blob/master/Game%20of%20Life/Game%20of%20Life/main.cpp>