

Game of Life

Vladislav Sobolev

December 2017

1 Simple solution

В начале я написал тривиальное решение задачи (без многопоточности). В этом решении я использую 2 массива размером $(n + 2) * (n + 2)$: *nowState* и *futureState*. Размер увеличен для того, чтобы обработка крайних значений поля совпадала с обработкой остальных полей. Сначала я заполняю массив *nowState* из условия (края соответственными противоположенными значениями). Затем, в цикле по количеству шагов заполняю массив *futureState* из соображений, что клетка станет живой, если у неё 3 живых соседа, или она была живой, и у неё 2 живых соседа. В том же цикле дописываю края, и в конце присваиваю *nowState* значение *futureState*.

2 MultiThread solution

Многопоточность я использовал следующим образом: я ввёл её при заполнении массива *futureState* горизонтально, т.е. разбил поле на блоки из какого-то количества строк, и каждый поток заполнял свой блок.

3 Results

Тестировал задачу на ноутбуке с процессором i7-7500U (2 ядра, 4 потока). Результаты представлены в таблице.

Тест/Тип решения	Тривиальное	1 поток	2 потока	3 потока	4 потока
10	21 ms	35ms	19 ms	16 ms	15 ms
100	52 ms	46 ms	39 ms	44 ms	50 ms
1000	1 s 207 ms	1s 184 ms	951 ms	865 ms	819 ms

Как можно видеть использование четырёх потоков сократило время работы на треть на тесте $n = 1000$.