

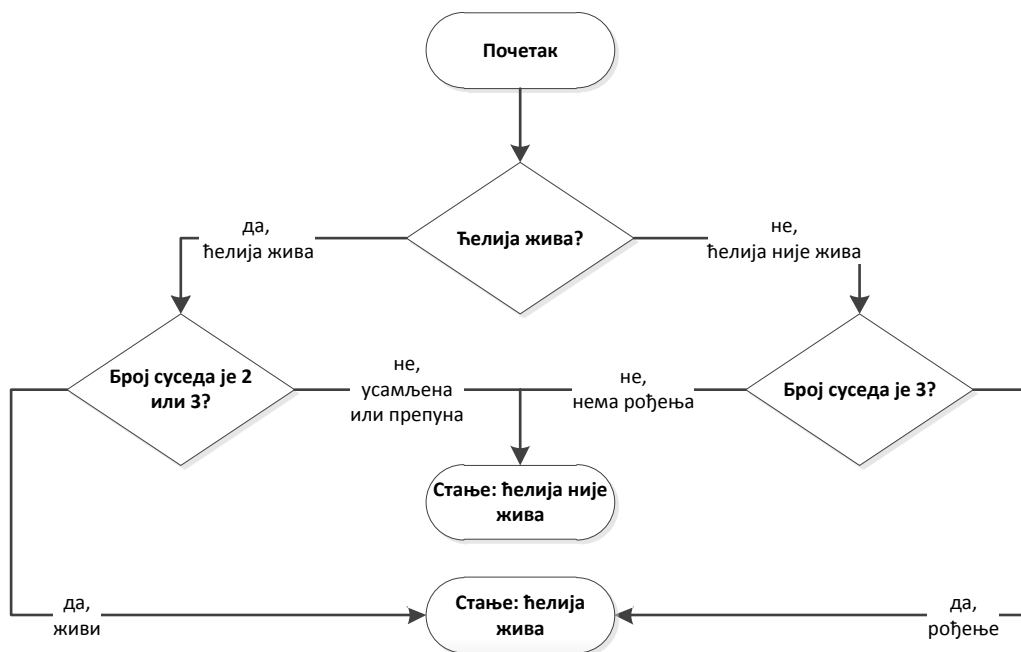
CILK SPAWN-SYNC И HYPEROBJECTS – ПРИМЕР СА „ИГРОМ ЖИВОТА“

Игра живота

„Игра живота“ је математичка игра коју је осмислио британски математичар Џон Хортон Конвеј (John Horton Conway) 1970. године. Игра се састоји од бесконачне квадратне табле са ћелијама, то јест пољима, као код шаховске табле. Свака ћелија је у интеракцији са ћелијама које је окружују (суседима) и може бити у једном од два стања: мртва ћелија или жива ћелија. За ову игру нису потребни играчи већ само почетно стање табле. Игра се одвија у циклусима у којима се примењују следећа правила:

1. Свака жива ћелија са мање од 2 жива суседа умире наредном циклусу
2. Свака жива ћелија са 2 или 3 жива суседа живи и у наредном циклусу
3. Свака жива ћелија са више од 3 жива суседа умире у наредном циклусу
4. Свака мртва ћелија са 3 жива суседа постаје жива у наредном циклусу

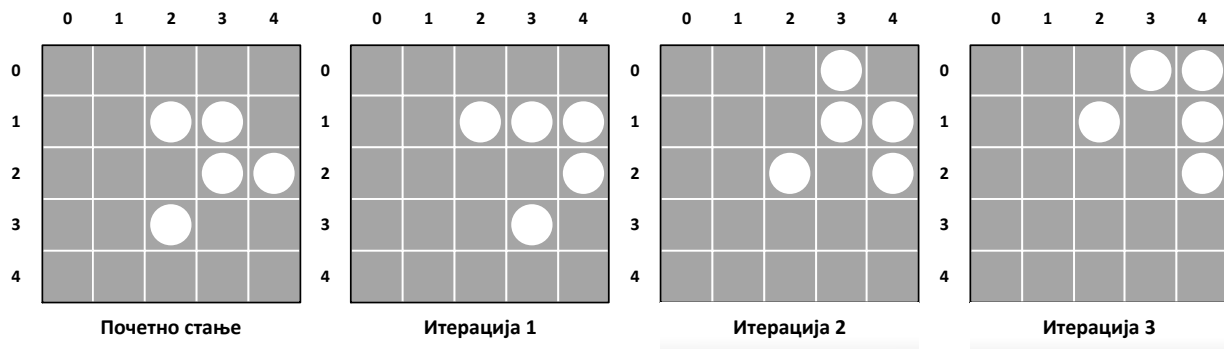
На следећој слици дат је алгоритам за игру живота који се извршава за сваку ћелију.



Слика 1: Алгоритам за игру живота за сваку ћелију

Пример

На следећој слици дато је почетно стање и 3 итерације игре живота. У наставку следи опис сваке итерације.



Слика 2: Пример игре живота

Итерација 1

Ред	Колона	Претходно стање	Ново стање	Правило
1	2	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
1	3	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
1	4	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.
2	3	жива	Мртва	Правило 3: Жива ћелија умире ако има 3 жива суседа.
2	4	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
3	2	жива	Мртва	Правило 1: Ћелија умире ако има мање од 2 жива суседа.
3	3	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.

Итерација 2

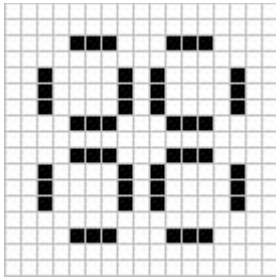
Ред	Колона	Претходно стање	Ново стање	Правило
0	3	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.
1	2	жива	Мртва	Правило 1: Ћелија умире ако има мање од 2 жива суседа.
1	3	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
1	4	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
2	2	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.
2	4	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
3	3	жива	Мртва	Правило 1: Ћелија умире ако има мање од 2 жива суседа.

Итерација 3

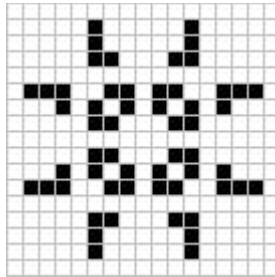
Ред	Колона	Претходно стање	Ново стање	Правило
0	3	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
0	4	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.
1	2	Мртва	жива	Правило 4: Ћелија се рађа ако има 3 жива суседа.
1	3	жива	Мртва	Правило 3: Жива ћелија умире ако има 3 жива суседа.
1	4	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа
2	2	жива	Мртва	Правило 1: Ћелија умире ако има мање од 2 жива суседа.
2	4	жива	жива	Правило 2: Ћелија је жива ако има 2 или 3 жива суседа

ЗАДАЦИ

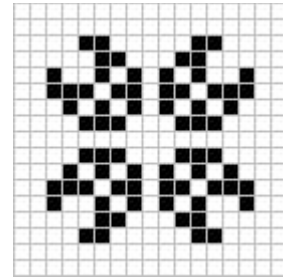
- 1) У приложеном референтном (серијском) програму потребно је имплементирати функцију за израчунавање броја суседа ћелије која је одређена произвољним координатама (*getNeighbourSum* функција).
- 2) Проверити исправност рада програма поређењем добијених решења у прве три итерације са решењима приказаним на сликама 3, 4 и 5 (доле).
- 3) Урадити паралелизацију референтног програма употребом CILK SPAWN-SYNC конструкције.



Слика 3: Почетно стање, итерација 3



Слика 4: Итерација 1



Слика 5: Итерација 2

Напомена: немојте се устручавати да по потреби направите додатне помоћне функције