

2006 Paper 1 Question 10

Programming in Java

The “Game of Life” is played on a board of square cells. Each cell is either “live” or “dead”. Initially most cells are dead, but a seeding pattern of live ones is set up. Each square cell has eight immediate neighbours (North, South, East, West, and four diagonal ones). At each time step all cells transform simultaneously. If a cell is dead, it becomes alive if it had (just before this time step) exactly three live neighbours. If it is alive, it becomes dead unless it has two or three live neighbours. In this question, locations beyond a 1000×1000 board are to be treated as permanently dead.

- (a) Show how to set up a simple Java 2-dimensional array of `boolean` values to represent a Life Board, with all cells initially “dead”. [2 marks]
- (b) For a location (i, j) on the board, give code that will decide whether the next state of that cell should be alive or dead. Make it clear how your code copes if the cell is at the boundary of the board. [4 marks]
- (c) Referring to part (b), write code that takes one board representing the current state of the game and fills in a second board-array with the state arrived at after one time step. What would happen if instead of using two arrays you wrote the new cell state directly back, using just a single copy of the board? [3 marks]
- (d) Re-work your solution to part (c) so that you can perform a time step using just one board. You may need to use a 1000-element vector to store information in a way that makes the update safe. [7 marks]
- (e) All the code you have written so far uses an array of `boolean` values. Some programmers would instead use an array of `int` values and treat each of the 32 bits in each `int` as giving the status of a cell. Suppose you have a 2-dimensional array of integers of size 1024 by 32 (that size is chosen so the array of integers may be viewed as a 1024 by 1024 array of bits): give code to retrieve a bit from a given position (i, j) . [4 marks]