CX 4010 / CSE 6010 Assignment 1: Modeling Forest Fires

Due Dates:

- Due: 10:00 AM, Friday, September 9, 2016
- Revision (optional): 11:55 PM, Monday September 12, 2016

1. Forest Fire Simulation

The objective of this assignment is to write a program to simulate the growth and decline (due to fires) of a forest over time. Here, the forest occupies an NxN checkboard-like two-dimensional grid. Each cell in the checkerboard is in one of two states: *occupied* means the square is occupied by a tree, and *empty* means the square contains no tree. Each cell, except those along the edges of the grid, has 4 neighbors: one to the north, south, east, and west.

The simulation will operate in a time-stepped fashion where the simulation is initialized at time 0 to an empty (no trees) state, and the new state of the grid is computed for time *1*, *2, 3*, etc. Within each time step the forest first goes through a growth phase where new trees are grown, followed by a fire phase where trees burn down due to fires. Specifically, the state at each time step is updated according to the following rules:

- *Growth phase*. If a cell is empty at time t, new growth will occur in the cell with probability g, and the state of the cell becomes occupied; conversely, the cell will remain empty with probability $1.0 - g$. All cells of the grid are updated to account for this new growth. Cells that already contain a tree remain in the same state (occupied).
- *Fire phase*. After the growth phase of this time step has been completed, fires may arise due to lighting strikes. If a cell contains a tree a lightning strike occurs in that cell with probability f causing the tree to catch fire; the state of the cell changes to empty. Further, whenever a fire occurs in a cell it immediately spreads to its neighboring cells that also contain a tree, and they too also become empty. This process continues, with the fire spreading to the neighbors of those neighbors, and so on, until the fire finally burns itself out due to a lack of additional occupied cells to which the fire can spread.
- After the state of all cells in the grid have been updated, advance to the next time step, and repeat the above growth and fire phases for the new time step.

We are interested in the average size (i.e., number of trees) in the forest as g and f are varied. For example, we might be interested in adjusting g by planting additional trees to help ensure the forest maintains a certain size, i.e., tree population. Define P as the average number of occupied cells at the end of each time step. Your simulation should compute the average value of P, computed over the length of the simulation run, and output this result.

The model described above is referred to as a cellular automaton. Cellular automata are widely used in the sciences, engineering, and social sciences to model a variety of different types of systems and phenomena.

2. Assignment (CX 4010 and CSE 6010)

To complete this assignment, develop a simulation of the system described above using the C programming language. Simulate a 100x100 grid. However, your program should be written so that the grid size can be easily changed. Similarly, the growth parameter g and fire probability f should be defined so they can be easily modified in your simulation. g and f remain the same throughout the simulation run.

In order to check that your code is working properly, write a C function that prints the state of the forest after each time step as well as any other information that you think will be useful to convince us that your code works properly. Embed calls to the function in your code to illustrate the operation of the simulator over a few time steps.

Once the simulation has been completed, complete a set of experiments with the goal of understanding how the forest size *P* will vary as *g* and *f* are changed. Specifically, create a graph with *P* on the vertical axis and *g* on the horizontal axis. Plot three curves with *f* equal to 0.01, 0.02, and 0.05.

Write up a brief report documenting your work and the results you obtained. This report should include evidence your program works correctly, e.g., by including suitable output and explanation.

Give a brief (a paragraph or two) explanation of the results computed by the simulation. In particular, describe what is going on with the forest as *g* is varied. What affect, if any, does *f* have? Be concise.

Turn your report and software in as a single zip file. Your software must be well documented and include comments so the code is easy to understand. You should include a README file with instructions on how to compile and run your program on the jinx cluster. Although this is not really necessary due to the relatively simple nature of this program, you should get into the habit of including such documentation with the software you develop.

Suggestion: The web is an excellent resource to answer specific questions on C. We encourage you to use it, but be careful not to utilize code copied directly from the web.

3. CSE 6010 Students

In addition to the above, conduct a literature survey to determine what results you would expect to obtain from this simulation model. Include citations to relevant literature as appropriate. Do your results agree with other results reported in the literature? Explain why or why not. Are there other applications where these results are relevant? While we are not expecting a fully comprehensive literature search, you should aim for a page or two of text with perhaps four or five related works cited.

4. CX 4010 Students (Extra Credit)

Complete the literature survey required for CSE 6010 students described above. If you complete the extra credit, we will note this in your assignment grade. At the end of the semester, if you are on the borderline between two letter grades, we will give you the higher grade if you have completed extra credit tasks throughout the semester.

5. Reminder: Collaboration Policy

A reminder you must adhere to the Georgia Tech honor code, and the collaboration policy stated in the course syllabus. Specifically, you are encouraged to discuss the problem and possible solutions with other students (as well as the TA/instructor), however, all code that you turn in must be completely your own work. Disseminating your code to other students is strictly prohibited. Further, downloading code from the web or other sources other than examples provided in class for use in this assignment is also prohibited.