

COMP 1771 – Fall 2008

HW#4

Instructor:

Prof. Matt Rutherford – mjr@cs.du.edu

Due by:

10:00am on October 23, 2008

Homework Policies

As stated in the class syllabus:

Homework assignments and labs may be discussed in a general fashion with other students. You should not discuss specific solutions or code. You should NEVER copy assignments that have been written by another student or allow another student to copy your assignments. If any of your work includes ideas or quotes from a book, paper, or web site, you should clearly cite the original source.

Your code should be emailed as attachments to me at mjr@cs.du.edu before 10am on October 23, 2008. Please use the subject: "COMP 1771 Homework #4".

Part 1: Self-Avoiding Random Walks

Adapted from Introduction to Programming in Java, Sedgewick and Wayne, Section 1.4 Example on Self-Avoiding Walks

Suppose that you leave your dog in the middle of a large city whose streets form a familiar grid pattern. We assume there are N north-south streets and N east-west streets all regularly spaced and fully intersecting in a pattern known as a *lattice*. Trying to escape the city, the dog makes a random choice of which way to go at an intersection, but knows by scent to avoid visiting any place previously visited. Unfortunately, it is possible for the dog to get stuck in a dead end where there is no choice but to revisit some intersection. What is the chance that this will happen?

Write a program `RandomWalk` that reads in the number N from the user and simulates (10000 times) the situation above with the dog starting at the intersection in the center of the city. Compute the probability of the dog winding up in a dead end.

HINT: model the intersections of streets with a 2-dimensional array of booleans.

Part 2: Crazy Bug

Consider a bug that is sitting in the middle of a grid that is initially painted black. The bug can move either up, down, left or right according to the following rules:

- If the current square is black, turn right 90-degrees, flip the color of the square, move forward one unit
- If the current square is white, turn left 90-degrees, flip the color of the square, move forward one unit

For this assignment, write an Applet class called `CrazyBug` that displays the board the bug is moving on (the dimensions of the board will be retrieved via the Applet's `getHeight()` and `getWidth()` and each pixel will be one grid square).

Initially, the bug will be in the middle of the screen with the entire grid painted black. The bug moves according to the rules described above and changing the colors of squares in the grid (i.e., drawing filled rectangles of 1x1 pixels). Return from the `paint(Graphics)` method when bug goes off the grid or after 100,000 iterations, whichever comes first.

Grading

Module	Percentage
Part 1: Self-Avoiding Random Walk	50%
Part 2: Crazy Bug	50%