CS 577: Introduction to Algorithms **Spring 2015**

Programming Project 1: Random Walks

Goal: To use empirical results to understand random walks on 1 and 2 dimensional arrays

Homework substitution: This programming project may be substituted for the written portion of Hw 1.

Due Date: 4pm Thursday March 26th (that's the Thursday just before spring break.)

Part 1. Random walks on 1-dimensional arrays.

Implementation. Implement a random walk on a 1-dimensional array. Initially assume that the array has 100 entries and begin the walk in position 50. But declare these values so that they can be easily changed. Use a random number generator to simulate the flip of a fair coin and walk toward the beginning of the array for tails and toward the end for heads. The walk stops if either end is reached. Use a variable t (time) to keep track of the number of coin flips.

Things to study:

1) As a function of t, the number of coin flips, what is the probability that the walk reaches an end of the array? That is, as a function of t what is the probability that the walk terminates? Plot this as a graph. Since you are running simulations, you won't get the same result each time. How many simulations did you need to run to make a reliable prediction? What happens if you increase the array size, or change the start point?

2) How many times are array cells visited before the walk ends? Run the simulation until the walk reaches an end of the array. For each array cell tabulate the number of times it is visited. Plot a histogram of your results. Do this in 2 ways: First, plot the number of cells that were visited 0 times, 1 time, 2 times, …. Second, let the x-axis simply range over the number of array locations (100) and plot on the y-axis the number of times that cell was visited.

Part 2. Random walks on 2-dimensional arrays.

Implementation. The implementation should use an nxn array with n easily changeable. The walk should start at an array location that is a variable location (i,j) within the array.

Things to study:

1) For a fixed value of n (eg, n=50), how large does t need to be for half the walks to reach an edge? Consider the start point (25,25). Plot the probability of stopping in t steps as a function of t.

2) When the walk terminates, how many cells have been visited?
3) When the walk terminates what is the distribution for the number of times that individual cells are visited?
4) If you change the rules and when a walk reaches the boundary, it "bounces back" into the array, how many steps does it take for the walk to visit all the cells? When all cells have been visited what is the distribution for the number of times that individual cells have been visited?

Things to turn-in.

Your Code:

1) A copy of your code for Parts 1 and 2. There is a dropbox set up on the Learn@UW site for Programming Project 1. You can simply drop you code there. One copy per group is fine.

A Written Report:

2) Graphs or plots for each of the 8 questions that we have asked you to study.
3) For each graph/plot write two paragraphs: the first explaining the way you conducted the simulations and the second explaining what you concluded from your simulations. This report should be printed and turned in during lecture or placed in the box on the door CS 4382.