COMP 1771 – Fall 2008
Class #11: Lab Exercises

Instructor:
Prof. Matt Rutherford – `mjr@cs.du.edu`

October 14, 2008

Hadamard Matrix

Adapted from Introduction to Programming in Java, Sedgewick and Wayne, Excercise 1.4.26

The $N \times N$ Hadamard matrix $H(N)$ is a boolean matrix with the remarkable property that any two rows differ in exactly $N/2$ entries. $H(1)$ is a 1×1 matrix with the single entry `true`, and for $N > 1$, $H(2N)$ is obtained by aligning four copies of $H(N)$ in a large square, and then inverting all of the entries in the lower-right $N \times N$ copy, as shown in the following examples:

$H(1)$

T

$H(2)$

T	T
T	F

$H(4)$

T	T	T	T
T	F	T	F
T	T	F	F
T	F	F	T

Write a program that reads N from the user and prints out $H(N)$. You may assume N is a power of 2.

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. Unfortulately, 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 that reads in the number N from the user and simulates 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.