Random Walk on a Hypercube

CS 2104 Homework Assignment 7

October 21, 2008

50 Points

The Problem. A random walk can be defined on a graph just as easily as we previously defined it on the integers. For this assignment, we will use the d-dimensional hypercube as the graph for the random walk. For $d \geq 1$, the *d-dimensional hypercube* is the graph with nodes that are bit strings (0's and 1's) of length d and an edge between u and v if u and v differ in exactly one bit. For a concrete example, see Homework Assignment 4.

For a node u of the cube, we can write $u = b_1 b_2 \cdots b_d$, a string of d bits. Its d neighbors are $b_1 b_2 \cdots \overline{b_i} \cdots b_d$, where $\overline{b_i}$ is the complementary bit to b_i. For example, if $d = 4$ and $u = 1011$, then its neighbors are 0011, 1111, 1001, and 1010.

Fix $d \geq 1$. For purposes of defining a random walk in the d-dimensional hypercube, the particle starts in some initial state S_0. If, after $i \geq 0$ steps, the particle is in state $S_i = b_1 b_2 \cdots b_d$, then, in step $i + 1$, it moves to any one of its d neighbors with equal probability $1/d$. If $d = 4$ and $S_0 = 0000$, then a possible sequence of states for the particle is

$$0000, 0010, 0110, 0100, 0101, 1101, 1111.$$

For every node $b_1 b_2 \cdots b_d$, there is an opposite "corner" $\overline{b_1 b_2} \cdots \overline{b_d}$, its *antipode*. A random walk will eventually travel from any node to its antipode. The number of steps for a random walk to travel from S_0 to its antipode the first time is called the *antipode time* of the random walk. For example, the antipode time for the above random walk is the number of steps from the first 0000 to the first 1111, which is 6. The average antipode time for the d-dimensional hypercube is T_d.

The Assignment. This assignment is to be done by the two assigned partners as a unit. The assignment is to write a program that will simulate a certain number of random walks on a d-dimensional hypercube for a given number of steps and that will estimate the average antipode time for the random walks. The program will be written in Java, C, or C++ as a single source file named `rwh.java` (Java), `rwh.c` (C), or `rwh.cpp` (C++).

The parameters for a simulation come from standard input as a single line of parameters, consisting of (1) d; (2) the initial state S_0; and (3) the number of random walks to simulate. For example, the parameter line

```
4 0010 2
```

specifies random walk simulations starting at $S_0 = 0010$ in the 4-dimensional hypercube, repeated 2 times. You are guaranteed that $1 \leq d \leq 10$.

The output of the simulation goes to standard output. The format of the output is the same as in Homework Assignment 4.

Using at least 100 repetitions for each run, estimate T_d for $1 \leq d \leq 10$. In a separate text file called `results.txt`, make a table containing your estimated T_d values. Can you take the sequence of T_d values and develop a function that approximates them? If so, then include the function in `results.txt`.

Submission. The submission for this assignment must be the source file for your program, plus `results.txt`. Each partnership uploads a single archive file (see below). The source file should be clearly commented and include the names of both partners. Put your source and `results.txt` files in either a `tar` or `zip` archive file. See the Programming Assignment Guidelines on the class home page for more relevant details. Your `tar` or `zip` file must be uploaded to Blackboard by 11:00 PM on Tuesday, October 21.