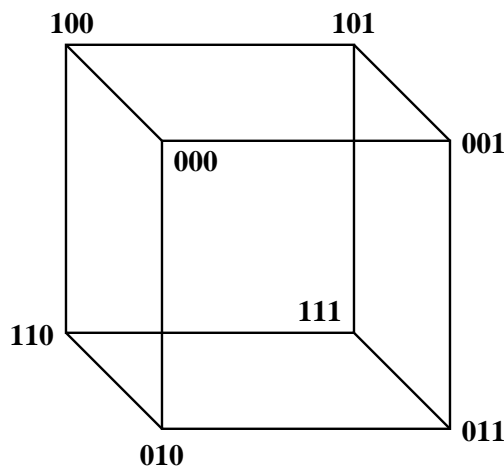


Random Walk on a Cube

CS 2104 Homework Assignment 3. Group work.

The Set-up 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. As a concrete example, choose $d = 3$. We get the *cube graph*, shown here:



For a node u of the cube, we can write $u = b_1b_2b_3$, a string of three bits. Its neighbors are $\overline{b_1}b_2b_3$, $b_1\overline{b_2}b_3$, $b_1b_2\overline{b_3}$, where $\overline{b_i}$ is the complementary bit to b_i . For example, if $u = 101$, then its neighbors are 001, 111, and 100.

For purposes of defining a random walk, the particle starts in some initial state S_0 . If, after $i \geq 0$ steps, the particle is in state $S_i = b_1b_2b_3$, then in step $i + 1$, it moves to state $\overline{b_1}b_2b_3$ with probability p ; to state $b_1\overline{b_2}b_3$ with probability q ; and to state $b_1b_2\overline{b_3}$ with probability r . It cannot stand still, so $p + q + r = 1$. Assume that $0 < p < 1$, $0 < q < 1$, and $0 < r < 1$. If $S_0 = 000$, then a possible sequence of states (trajectory) for the particle is

000, 001, 101, 001, 011, 010, 000, 100, 101, 100, 110, 111, 101, 111, 011.

For every node $b_1b_2b_3$, there is an opposite “corner” $\overline{b_1}\overline{b_2}\overline{b_3}$, 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 000 to the first 111, which is 11.

The Assignment. This assignment is to be done by the assigned group partners as a unit. It consists of two parts: Problem I (30 points) and Problem II (30 points).

Problem I The first problem of the assignment is to write a program that will simulate a certain number of random walks on a cube 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, C++, Matlab or Mathematica as a single source file named `rw.c.java` (Java), `rw.c` (C), or `rw.cpp` (C++).

The parameters for a simulation come from standard input as a single line of parameters, consisting of (1) the initial state S_0 ; (2) the value of p ; (3) the value of q ; (4) the value of r ; and (5) the number of random walks to simulate. For example, the parameter line

```
001 0.25 0.35 0.40 2
```

specifies random walk simulations starting at $S_0 = 001$, with $p = 0.25$, $q = 0.35$, and $r = 0.40$, and repeated 2 times.

The output of the simulation goes to standard output. First, S_0 is printed. As the simulation proceeds, each new state is printed, one state per line. The first step that the antipode of S_0 is reached, the simulation of that random walk is terminated, a blank line is output, and a new simulation beginning at S_0 is started. This is repeated the specified number of times. Finally, the estimated average antipode time is printed (as below). Here is a sample output that uses the previous sample parameters.

```
001
101
111
011
010
110
```

```
001
000
100
000
010
011
111
110
```

```
Average antipode time is 6.0
```

Submission. The submission for this assignment must be the source file for your program. Each partnership uploads a single source file. The source file should be clearly commented and include the names of all partners. As is always the case with group work, indicate clearly who did what, including who was the group leader.

Problem II Now, let's modify the above code and use it to answer a real life question. Suppose you have a particle in 3D space ($d=3$) that moves along a "random walk" trajectory specified by the rules you have just coded up. This is Brownian motion: the particle is being pushed about by jolts of thermal energy. Or you can think of it as a walk in the woods where you don't have any sense of direction. *Of course, now you need to move beyond the assumption that the particle is confined to cube: it can move anywhere on a lattice of integer points.* Each individual move obeys the same rule as before: only moves to any of its 6 nearest neighbors along the edge are allowed. That is suppose you start at $(0, 0, 0)$. The six neighbors are $(-1, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, -1, 0)$, $(0, 0, 1)$, $(0, 0, -1)$ The following trajectory is possible:

000, 001, 002, 012, 013.

while this one is not possible:

000, 002, 100.

(That is you *can* get to (002) from (000), but this will require at least two successive moves.)

Here is the question: how does the *displacement* ΔL of the particle from its starting point depend on time, on average, for large enough times? You can compute the displacement as the "normal" (Euclidean) distance in space between the start and end points of the trajectory. What you will need to do to answer the question is to accumulate a fairly large, $N \gg 1$, number of individual trajectories with different number of steps t in each. (1 step = 1 unit of time). For simplicity you can assume that they all start at $(0, 0, 0)$. For simplicity, assume equal probability of move in any direction: $p = q = r = 1/3$. Explore several values of t in the interval $10 \leq t \leq 100$. Prepare to expand the interval if the results are inconclusive. The more points you collect, the smaller the "noise". Do not hesitate to test the same values of t more than once. For each trajectory compute the distance ΔL between its start and finish point. Then put all your $(\Delta L, t)$ on a cumulative plot ΔL vs. t , where t is the total number of steps in each trajectory. Answer the following question: which law best describes your data points: $\Delta L \sim \sqrt{t}$, $\Delta L \sim t$, or $\Delta L \sim t^2$? Present both the graph and your conclusion. If you have time, explore how the answer may depend on the dimension of your space d (extra credit).

Submission. The submission for this assignment must be a one page write-up of your results, including a brief description of trajectories you ran (how many, what length, etc), the graph described above, and your conclusions. Each partnership uploads a single document file. As is always the case with group work, indicate clearly who did what.

Random Numbers. To complete the program, you will need a source of random numbers. See Homework Assignment 2 for the details. Suppose that the current state is $S_i = b_1 b_2 b_3$ and that you have a random number x uniformly distributed in $[0, 1]$, so

$0 \leq x \leq 1$. If $0 \leq x < p$, then the next state is $\overline{b_1}b_2b_3$. If $p \leq x < p+q$, then the next state is $b_1\overline{b_2}b_3$. If $p+q \leq x \leq 1$, then the next state is $b_1b_2\overline{b_3}$.