Homework 3

Due: Feb 21, 2019 at 2:30pm

Please follow these guidelines when writing up your CS 155 homeworks.

A major goal of this class is that you learn how to mathematically analyze probabilistic processes and events. Hence, make sure your proofs are rigorous, i.e., that every step is adequately justified. Try to keep your proofs as simple and as concise as possible, keeping in mind that you will potentially have to examine multiple proof strategies in order to achieve this. Make sure to CLEARLY STATE YOUR ASSUMPTIONS at the beginning of any solution, when necessary. As we are doing probabilistic analysis, we shall require that you clearly define the probability space you are working with and identify the events within it that you will analyze.

You are allowed to discuss these problems with other students, but you have to leave any meeting with other students without any written material about the problems, and the writeups must be done by yourself without help from others. If you have any questions regarding how these guidelines apply to a particular problem or what they mean in general, please email the TAs.

Problem 1

Exercise 4.22 in the textbook

Problem text:

Consider the bit-fixing routing algorithm for routing a permutation on the $n$-cube. Suppose that $n$ is even. Write each source node $s$ as the concatenation of two binary strings $a_s$ and $b_s$ each of length $n/2$. Let the destination of $s$’s packet be the concatenation of $b_s$ and $a_s$. Show that this permutation causes the bit-fixing routing algorithm to take $\Omega(\sqrt{N})$ steps.

Problem 2

Exercise 4.26 in the textbook

Problem text:

In this exercise, we design a randomized algorithm for the following packet routing problem. We are given a network that is an undirected connected
graph $G$, where nodes represent processors and the edges between the nodes represent wires. We are also given a set of $N$ packets to route. For each packet we are given a source node, a destination node, and the exact route (path in the graph) that the packet should take from the source to its destination. (We may assume that there are no loops in the path.) In each time step, at most one packet can traverse any single edge. A packet can wait at any node during any time step, and we assume unbounded queue sizes at each node.

A schedule for a set of packets specifies the timing for the movement of packets along their respective routes. That is, it specifies which packets should move and which should wait at each time step. Our goal is to produce a schedule for the packets that tries to minimize the total time and the maximum queue size needed to route all the packets to their destinations.

a. The dilation $d$ is the maximum distance traveled by any packet. The congestion $c$ is the maximum number of packets that must traverse a single edge during the entire course of the routing. Argue that the time required for any schedule should be at least $\Omega(c + d)$.

b. Consider the following unconstrained schedule, where many packets may traverse an edge during a single time step. Assign each packet an integral delay chosen randomly, independently, and uniformly from the interval $[1, \lceil \alpha c / \log(Nd) \rceil]$, where $\alpha$ is a constant. A packet that is assigned a delay of $x$ waits in its source node for $x$ time steps; then it moves on to its final destination through its specified route without ever stopping. Give an upper bound on the probability that more than $O(\log(Nd))$ packets use a particular edge $e$ at a particular time step $t$.

c. Again using the unconstrained schedule of part (b), show that the probability that more than $O(\log(Nd))$ packets pass through any edge at any time step is at most $1/(Nd)$ for a sufficiently large $\alpha$.

d. Use the unconstrained schedule to devise a simple randomized algorithm that, with high probability, produces a schedule of length $O(c + d \log(Nd))$ using queues of size $O(\log(Nd))$ and following the constraint that at most one packet crosses any single edge per time step.