Homework 7 – Due Thursday, March 15, 2018 at 4:45pm

Submit solutions to all problems on separate sheets. They will be graded by different people.

Page limit You can submit at most 1 sheet of paper per problem, even if the problem has multiple parts. If you submit a longer solution for some problem, only the first sheet of paper will be graded.

Reminder Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to the instructor if asked. You must also identify your collaborators and whether you gave help, received help, or worked something out together. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises Please practice on exercises in Chapter 4 of Mitzenmacher-Upfal.

Problems

1. (Random vectors) Consider $d$-dimensional vectors, where $d$ is a natural number. Recall that two vectors $a$ and $b$ are orthogonal if their dot product $a \cdot b = \sum_{i \in [d]} a_i b_i$ is zero. For a real number $\epsilon > 0$, two vectors are $\epsilon$-close to being orthogonal if their dot product is in the interval $[-\epsilon, \epsilon]$.

   (a) You pick two $d$-dimensional unit vectors independently at random by setting each coordinate uniformly to $1/\sqrt{d}$ or $-1/\sqrt{d}$. Give the best upper bound you can on the probability that the two vectors are not $1/10$-close to orthogonal.

   (b) In $d$ dimensions, at most $d$ vectors can be pairwise orthogonal. However, exponentially many (in $d$) vectors can be close to pairwise orthogonal. Moreover, a random collection of exponentially many vectors is likely to be close to pairwise orthogonal. Find as large $k$ as you can (as a function of $d$) such that $k$ unit vectors chosen independently as specified in part (a) are pairwise $1/10$-close to being orthogonal.

2. (Randomized Routing on the Hypercube) Recall the Randomized Routing Algorithm and its analysis from class. As part of the analysis, we stated a lemma that we did not have time to prove. In this problem, you will prove this lemma.

   (a) Consider each route in Phase 1 of the algorithm as a directed path from the source $x$ to the designation $z$. Prove that once two routes separate, they do not rejoin.

   (b) Does part (a) imply that, for any two packets $i$ and $j$, there is at most one node such that $i$ and $j$ are waiting in queue at that node at the same time step?

   (c) Consider any packet $i$. Let $p_i = (v_1, \ldots, v_k)$ be its path in phase 1. Let $S$ be the set of packets (other than $i$) whose routes pass through at least one edge of $p_i$. Recall that the delay of a packet is the number of time steps it waits in queues (in Phase 1). Show that the delay of packet $i$ is at most $|S|$.

   Hint: Use part (a).

   Guidelines: For $j \in [k-1]$, let $n_j$ be the number of packets entering $p_i$ for the first time at vertex $v_j$. Consider the time step $t$ at which the last packet from $S$ uses the edge $(v_j, v_{j+1})$. Define the lag at vertex $v_j$ to be the integer $t - j$ and denote it $\ell_j$. 

i. Argue that $\ell_1 \leq n_1$.

ii. Argue that $\ell_j \leq \ell_{j-1} + n_j$ for all $j \in \{2, \ldots, k\}$.

iii. Argue that the delay of packet $i$ is at most $|S|$.

Recall that, as part of our analysis of the Randomized Routing Algorithm, we defined a random variable $Y_e$ to be the number of packets whose routes in Phase 1 use edge $e$ of the hypercube. Next, you will calculate the expectation of $Y_e$ in a different way than we did in class.

(d) Let $L_i$ be the number of edges in the route of packet $i$. Argue that $\sum_e Y_e = \sum_i L_i$ for appropriately specified $e$ and $i$. (Make sure you specify what $e$ and $i$ we are summing over.)

(e) Find the expectation of the summation in part (d).

(f) Use part (e) to derive $E[Y_e]$.

3. (Randomized Quicksort) Exercise 4.21 from Mitzenmacher-Upfal. (Recall that we analyzed the expected running time of Quicksort in class. In this problem, you will work out a high probability statement about the running time.)