Discrete Structures	**Problem Set 8**
CS 280 Spring 2005	**Due Wednesday, April 6, 2005**

Reading: Rosen edition 5: Sections 5.1–2, or edition 4: Sections 4.4–4.5.

Kleinberg-Tardos: Algorithm Design: Sections 13.1, 13.3 and 13.12. Handout with the relevant section of the book is available from the racks in front of Upson 303.

Prelim II: The second prelim will be on Tuesday, April 12th, 7:30-9pm. We will have an **in class** review session on Monday, April 11th. The topics that are covered in the prelim are listed in blue on the Schedule (available from the course Web page. Practice questions will be available on April 6th.

(1) Consider a county in which $100,000$ people vote in an election. There are only two candidates on the ballot: a Democratic candidate (denoted D) and a Republican candidate (denoted R). As it happens, this county is heavily Democratic, so $80,000$ people go to the polls with the intention of voting for D and $20,000$ go to the polls with the intention of voting for R.

However, the layout of the ballot is a little confusing, so each voter, independently and with probability $\frac{1}{100}$, votes for the wrong candidate; i.e. the one that he or she *didn't* intend to vote for. (Remember that in this election, there are only two candidates on the ballot.)

Let X denote the random variable equal to the number of votes received by the Democratic candidate D, when the voting is conducted with this process of error. Determine the expected value of X, and give an explanation of your derivation of this value.

(2) Spam filters are working to decide if a message is likely to be spam based on simple properties, like its sender, subject, etc. The basic idea can be expressed as the following simple calculation.

Assume that of all messages arriving to Cornell, 40% is spam. Over all messages only 5% comes from a hotmail account, but of spam messages 10% is coming from hotmail. What is the probability that a message from hotmail to Cornell is spam, and what is the probability that a message to Cornell that is not from hotmail is spam? Express this as a conditional probability. What is the sample space, what are the events discussed in the problem, and explain the reason for the probability you stated.

(3) Assume in a city 60% of all voters are republicans and 40% are democrats (no independents for this problem). We know that 65% of democrats oppose military spending, and only 40% of the republican oppose military spending. What is the conditional probability that a randomly selected voter who opposes military spending is a democrat?

Hint: recall Bayes rule from class: for any two events A and B we have the following (where $\bar{A}$ is the complement of event A):

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\bar{A})\Pr(\bar{A})}.$$

You may want to use this rule, and consider the probability space of uniformly randomly selected voter in the city above. And the two events in question are A= democrats, $\bar{A}$= republicans, and B= voters opposing military spending.

(4) In Mondays class (and the first section of the handout) we saw a simple distributed protocol to solve a particular contention-resolution problem. Here is another setting in which randomization can help with contention-resolution, through the distributed construction of an independent set.

Suppose we have a system with n processes. Certain pairs of processes are in *conflict*, meaning that they both require access to a shared resource. In a given time interval, the goal is to schedule a large subset S of the processes to run — the rest will remain idle — so that no two conflicting processes are both in the scheduled set S. We'll call such a set S *conflict-free*.

We'd like a simple method for selecting a large conflict free set without centralized control: each process should communicate with only a small number of other processes, and then decide whether or not it should belong to the set S.

Suppose each process has exactly d other processes that it conflicts with.

(a) Consider the following simple protocol.

> Each process P_i independently flips a coin, H or T; it selects H with probability $\frac{1}{2}$ and selects T with probability $\frac{1}{2}$. It then decides to enter the set S if and only if it chooses H and each of the processes with which it is in conflict chooses T.

Note that the set S resulting from the execution of this protocol is conflict-free. Give a formula for the probability that a process enters S, and give a formula for the expected size of S in terms of n (the number of processes) and d (the number of conflicts per process). Briefly explain your answer.

(b) The choice of the probability $\frac{1}{2}$ in the protocol above was fairly arbitrary, and it's not clear that it should give the best system performance. A more general specification of the protocol would replace the probability $\frac{1}{2}$ by a parameter p between 0 and 1, as follows:

> Each process P_i independently flips a coin, H or T; it selects H with probability p and selects T with probability $(1-p)$. It then decides to enter the set S if and only if it chooses H and each of the processes with which it is in conflict chooses T.

In terms of n (the number of processes) and d (the number of conflicts per process) give a value of p so that the expected size of the resulting set S is as large as possible. Give a formula for the expected size of S when p is set to this optimal value.

(5) *Load-balancing algorithms* for parallel or distributed systems seek to spread out collections of computing jobs over multiple machines. In this way, no one machine becomes a "hot spot." If some kind of central coordination is possible, then the load can potentially

be spread out almost perfectly. But what if the jobs are coming from diverse sources that can't coordinate? One option is to assign them to machines at random, and hope that this randomization will work to prevent imbalances. Clearly this won't generally work as well as a perfectly centralized solution, but it can be quite effective. Here we try analyzing, in one particular model, the effectiveness of a randomized load balancing heuristic.

Suppose you have k machines, and k jobs show up for processing. Each job is assigned to one of the k machines independently at random (with each machine equally likely).

(a) what is the probability that a given machine j receives no jobs? Give a proof of your answer.

(b) What is a probability that a given machine j receives exactly 1 job. Give a proof of your answer.

(c) What is the probability that each machine gets exactly one job? Show that this probability converges to 0 as k goes to infinity.

(d) Let $N(k)$ be the expected number of machines that do not receive any jobs, so that $N(k)/k$ is the expected fraction of machines with nothing to do. What is the value of the limit $\lim_{k\to\infty} N(k)/k$? Give a proof of your answer. You may use the fact that $\lim_{k\to\infty}(1-\frac{1}{k})^k = \frac{1}{e}$ without proof.

(e) Suppose that machines are not able to queue up excess jobs, so if the random assignment of jobs to machines sends more than one job to a machine M, then M will do the first of the jobs it receives and reject the rest. Let $R(k)$ be the expected number of rejected jobs; so $R(k)/k$ is the expected fraction of rejected jobs. What is $\lim_{k\to\infty} R(k)/k$? Give a proof of your answer.

(6 optional) Suppose you're designing strategies for selling items on a popular auction Web site. Unlike other auction sites, this one uses a *one-pass auction*, in which each bid must be immediately (and irrevocably) accepted or refused. Specifically,

- First, a seller puts up an item for sale.
- Then buyers appear in sequence.
- When buyer i appears, he or she makes a bid $b_i > 0$.
- The seller must decide immediately whether to accept the bid or not. If the seller accepts the bid, the item is sold and all future buyers are turned away. If the seller rejects the bid, buyer i departs and the bid is withdrawn; and only then does the seller see any future buyers.

Suppose an item is offered for sale, and there are n buyers, each with a distinct bid. Suppose further that the buyers appear in a random order, and that the seller knows the number n of buyers. We'd like to design a *strategy* whereby the seller has a reasonable chance of accepting the highest of the n bids. By a "strategy," we mean a rule by which the seller decides whether to accept each presented bid, based only on the value of n and the sequence of bids seen so far.

For example, the seller could always accept the first bid presented. This results in the seller accepting the highest of the n bids with probability only $1/n$, since it requires the highest bid to be the first one presented.

Give a strategy under which the seller accepts the highest of the n bids with probability at least $1/4$, regardless of the value of n. (For simplicity, you are allowed to assume that n is an even number.) Prove that your strategy achieves this probabilistic guarantee.