Discrete Structures	**Problem Set 9**
CS 280 Spring 2005	**Due Wednesday, April 27, 2005**

Reading: Rosen edition 5: Sections 5.3 and 7.1, 7.5 and 8.1-3. Or from edition 4: Sections 4.5 and 6.1 6.5 and 7.1-3.

(1) We have shown in class Chebyshev's inequality that states that for every random variable X with expectation $E(X) = \mu$, variance $\nu = V(X)$, standard deviation $\sigma(X) = \sqrt{\nu}$, and any $k > 1$ we have the following bound on the probability that X deviates from μ by too much:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

(Note that in class we used strict inequalities in place of both inequalities. Chebyshev's inequality is valid in both forms.) Show that this bound is the strongest one can prove of this form. More formally, show that for any μ, and any $\sigma > 0$ and any $k > 1$, there is a random variable X with $E(X) = \mu$, $\sigma(X) = \sigma$, and $P(|X - \mu| \geq k\sigma) = \frac{1}{k^2}$.

(2) A way to improve the outcome of a random decision process is to repeat it multiple times, and take the majority. To model this consider the following scenario. Our decision process will be modeled as a biased coin, that is more likely to make the right decision, but can be wrong with say 40% probability. Concretely, assume we have a biased coin that lands on one side with 60% probability, and lands on the other with 40% probability. However, we don't know which is which.

(a) A simple way to decide which side is more likely is to toss to coin, and declare the side it lands on the "likely" side. What is the probability that this decision is correct? Explain why your answer is correct.

(b) Suppose instead we toss the coin three times independently, and declare the side it lands more often on, as the likely side. So for example, if we get H,T,T as a sequence, then we say that T is the likely side. What is the probability that this decision is correct? Explain why your answer is correct.

(c) Suppose instead we toss the coin $2k + 1$ times independently, and declare the side it lands more often on, as the likely side. Bound the probability that this decision is correct using the Chebyshev bound. Show that this probability goes to 1 as k gets large. Explain why your answer correct.

(3) In class we talked about the following model for frequency of down-loads from a Web server. Time is divided into n slots, and in each slot independently, with probability λ/n we get a down-load request. Note that for all values of $n > 0$ the expected number of request is $np = \lambda$. The probability that exactly k requests arrive is

$$f_n(k) = \binom{n}{k} (\frac{\lambda}{n})^k (1 - \frac{\lambda}{n})^{n-k}.$$

We also discussed that the limit of this value as n goes to infinity is $f(k) = e^{-\lambda}\frac{\lambda^k}{k!}$. The probability distribution we get this way is called the Poisson process. We have seen that the expected value of the number of down-loads in an hour is λ for all values $n > 0$ and also for the limit.

In this question we want to consider the same distribution, and ask what is the expected time between two down-load requests.

(a) Assume as before that time is divided by dividing each hour into n slots, and in each slot independently, with probability λ/n we get a down-load request. What is the probability that we need to wait t hours between two requests. Note that t hours are tn time slots and t does not have to be an integer, but you may want to assume here that nt is integer.

(b) Express the expected waiting time between two requests as function of n and λ.

(4) A number of *peer-to-peer systems* on the Internet are based in the following idea. If n users share files, and a new user $n + 1$ arrives, this user can down-load these files from any of the previous n users. One simple way to implement this is that each new user selects one of the previously arrived users independently at random, and down-loads the file from that user. For this problem, we will assume that users never leave. So the system starts with a single 1st user that has the file. When the 2nd user arrives, he/she down-loads the file from the 1st user. When the 3rd user arrives, he/she randomly chooses between the previous users (1st and 2nd) and down-loads the file from the selected user, etc. One issue with this simple scheme is that it is a bit unfair: later arriving users don't have to serve as many down-loads as the earlier ones. To quantify, this let $1, 2, \ldots, n$ denote the sequence of n users in the order they arrive.

(a) What is the expected number of down-loads that user j will have to serve. Express this expectation as a function of j and n. Explain your answer.

(b) Part (a) makes precise a sense in which the nodes that arrive early serve an "unfair" share of the down-loads in the network. Another way to quantify the imbalance is to consider users that never serve any down-loads. What is the probability that user j never serves any down-loads? express this probability in terms of n and j.

(c) Give a formula for the expected number of users who never serve any down-load requests. For full credit, you need to simplify your expression, so it does not involve long summations or products, or $\sum$ signs, etc.

(5) Consider a simple undirected graph on n nodes. Recall that the degree of a node is the number of adjacent edges. Is it true that all graphs have two nodes $u \neq v$ that have equal degree $deg(u) = deg(v)$? Prove your claim either by proving that two such nodes must always exists, or by providing a counter example. You may want to solve this problem only after Friday.

(6) [from Kleinberg-Tardos] Some friends of yours work on wireless networks, and they're currently studying the properties of a network of n mobile devices. As the devices move around (really, as their human owners move around), they define a graph at any point in time as follows: there is a node representing each of the n devices, and there is an edge between device i and device j if the physical locations of i and j are no more than 500 meters apart. (If so, we say that i and j are "in range" of each other.)

They'd like it to be the case that the network of devices is connected at all times, and so they've constrained the motion of the devices to satisfy the following property: at all times, each device i is within 500 meters of at least $n/2$ of the other devices. (We'll assume n is an even number.) What they'd like to know is: Does this property by itself guarantee that the network will remain connected?

Here's a concrete way to formulate the question as a claim about graphs:

> *Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected.*

Decide whether you think the claim is true or false, and give a proof of either the claim or its negation. You may want to solve this problem only after Friday.

(7) A set of $2k + 1$ people would like to agree flip a random coin. One way to do this would be to ask one person A to flip a coin, and announce to the rest of them the outcome. Unfortunately, one of the $2k + 1$ people is a cheater, and instead of flipping a coin, he would just choose the outcome that works best for him. Also, we don't know who the cheater is. You can think of the problem as follows. We ask all of the $2k+1$ people to flip a random coin. $2k$ of them will flip unbiased coins independently, but 1 person will just choose whatever outcome he needs. If it helps, you can ask the people to flip another coin, again all but one of them would flip a new coin, again unbiased random, but the one cheater may cheat again (and does not have to cheat the same way).

If you select one of the tossed coins as the "random" outcome, there is always a small chance that the selected coin was "tossed" by the cheater, and hence it is not random.

Design a system with the following properties. You need to select outcome of a random coin, that is unbiased, and the cheater cannot effect the outcome. To do so, you may ask the people to flip a sequence of coins, but the expected number of coin flips should not be too high. Explain how your system works, and why it has the claimed property. You may use the fact that the probability that we get k heads when flipping $2k$ independent random coins is very small.