Algorithms	**Problem Set 6**
CS 6820 Fall 2009	**Due November 30, 2009**

Recall the general instructions for handing in homework:

- If possible, please typeset the homework (i.e. format your solutions as an electronic file using latex or Word with mathematical notation).
- Homework solutions done electronically can be handed in by directly uploading them to CMS. Please mail Ashwin (`ashwin85@cs.cornell.edu`) if you have any trouble with this.

(1) *(KT Exercise 13.4)* A number of *peer-to-peer systems* on the Internet are based on *overlay networks*: rather than using the physical Internet topology as the network on which to perform computation, these systems run protocols by which nodes choose collections of virtual "neighbors" so as to define a higher-level graph whose structure may bear little or no relation to the underlying physical network. Such an overlay network is then used for sharing data and services, and it can be extremely flexible compared with a physical network, which is hard to modify in real-time to adapt to changing conditions.

Peer-to-peer networks tend to grow through the arrival of new participants, who join by linking into the existing structure. This growth process has an intrinsic effect on the characteristics of the overall network. Recently, people have investigated simple abstract models for network growth that might provide insight into the way such processes behave, at a qualitative level, in real networks.

Here's a simple example of such a model. The system begins with a single node v_1. Nodes then join one at a time; as each node joins, it executes in a protocol whereby it forms a directed link to a single other node chosen uniformly at random from those already in the system. More concretely, if the system already contains nodes $v_1, v_2, \ldots, v_{k-1}$ and node v_k wishes to join, it randomly selects one of $v_1, v_2, \ldots, v_{k-1}$ and links to this node.

Suppose we run this process until we have a system consisting of nodes $v_1, v_2, \ldots, v_n$; the random process described above will produce a directed network in which each node other than v_1 has exactly one out-going edge. On the other hand, a node may have multiple in-coming links, or none at all. The in-coming links to a node v_j reflect all the other nodes whose access into the system is via v_j; so if v_j has many in-coming links, this can place a large load on it. To keep the system load-balanced, then, we'd like all nodes to have a roughly comparable number of in-coming links, but that's unlike to happen here, since nodes that join earlier in the process are likely to have more in-coming links than nodes that join later. Let's try to quantify this imbalance as follows.

(a) Given the random process described above, what is the expected number of in-coming links to node v_j in the resulting network? Give an exact formula in terms of n and j, and also try to express this quantity asymptotically (via an expression without large summations) using $\Theta(\cdot)$ notation.

(b) Part (a) makes precise a sense in which the nodes that arrive early carry an "unfair" share of the connections in the network. Another way to quantify the imbalance is to observe that, in a run of this random process, we expect many nodes to end up with no in-coming links.

Give a formula for the expected number of nodes with no in-coming links in a network grown randomly according to this model. (Again, specify this using an expression without large summations.)

(2) *(KT Exercise 13.15)* Suppose you are presented with a very large set S of real numbers, and you'd like to approximate the median of these numbers by sampling. You may assume all the numbers in S are distinct. Let $n = |S|$; we will say that a number x is an *ε-approximate median* of S if at least $(\frac{1}{2} - \varepsilon)n$ numbers in S are less than x, and at least $(\frac{1}{2} - \varepsilon)n$ numbers in S are greater than x.

Consider an algorithm that works as follows. You select a subset $S' \subseteq S$ uniformly at random, compute the median of S', and return this as an approximate median of S. Show that there is an absolute constant c, independent of n, so that if you apply this algorithm with a sample S' of size c, then with probability at least .99, the number returned will be a (.05)-approximate median of S. (You may consider either the version of the algorithm that constructs S' by sampling with replacement, so that an element of S can be selected multiple times, or without replacement.)

(3) Consider a long, straight road, which we model as a line segment of length n. We drop a set of k sensors randomly on this road — so each lands in a location equal to a real number selected uniformly and independently from the interval $[0, n]$. We also have two sensors a^* and b^* that are placed deterministically at the start and end of the road — that is, a^* is placed at location 0 and b^* is placed at location n in the interval $[0, n]$. This gives us $k + 2$ sensors in total: k with uniform random locations, and two with deterministically chosen locations.

Now, each sensor has a transmitting range of 1, so it can communicate with any other sensor within a distance 1 of it. This means that the random placement of the sensors defines a random $(k + 2)$-node graph G, in which the nodes are the sensors, and we connect two by an edge if they can communicate with each other.

(a) Is the following claim true or false?

> *Claim: For some constant c (independent of n), and some function $f(n)$ such that $\lim_{n\to\infty} f(n) = 0$, if we place $k = cn \log n$ sensors, then G is connected with probability at least $1 - f(n)$.*

Give a proof for your answer.

(b) Is the following claim true or false?

> *Claim: For some constants c_1 and c_2 (independent of n), and some function $g(n)$ such that $\lim_{n\to\infty} g(n) = 0$, if we place $k = c_1 n \log n$ sensors, then with probability at least $1 - g(n)$, the graph G contains at least $c_2 \log n$ node-disjoint paths from a^* and b^* — that is, at least $c_2 \log n$ paths from a^* to b^* that mutually have no nodes in common.*

Give a proof for your answer.