Introduction to Algorithms	**Problem Set 4**
CS 482 Spring 2000	**Due March 10, 2000**

(1) Suppose you're looking at a flow network G with source s and sink t, and you want to be able to express something like the following intuitive notion: some nodes are clearly on the "source side" of the main bottlenecks; some nodes are clearly on the "sink side" of the main bottlenecks; and some nodes are in the middle. However, G can have many minimum cuts, so we have to be careful in how we try making this idea precise.

Here's one way to divide the nodes of G into three categories of this sort.

- We say a node v is *upstream* if for all minimum s-t cuts (A, B), we have $v \in A$ — that is, v lies on the source side of every minimum cut.
- We say a node v is *downstream* if for all minimum s-t cuts (A, B), we have $v \in B$ — that is, v lies on the sink side of every minimum cut.
- We say a node v is *central* if it is neither upstream nor downstream; there is at least one minimum s-t cut (A, B) for which $v \in A$, and at least one minimum s-t cut (A', B') for which $v \in B'$.

Give an algorithm that takes a flow network G, and classifies each of its nodes as being upstream, downstream, or central. The running time of your algorithm should be within in a constant factor of the time required to compute a *single* maximum flow; so, following the best bound from the course packet (Section 5.4), it should have running time at most $O(m^2 \log C)$.

(2) People like to claim that much of the economic potential in a company like *Yahoo!* is in the "eyeballs" — the simple fact that it gets millions of people looking at its pages every day. And further, by convincing people to register personal data with the site, it can show each user an extremely targeted advertisement whenever he or she visits the site, in a way that TV networks or magazines couldn't hope to match. So if the user has told *Yahoo!* that they're a 20-year old computer science major from Cornell University, the site can throw up a banner ad for apartments in Ithaca, NY; on the other hand, if they're a 50-year-old investment banker from Greenwich, Connecticut, the site can display a banner ad pitching Lincoln Town Cars instead.

But deciding on which ads to show to which people involves some serious computation behind the scenes. Suppose that the managers of a popular Web site have identified k distinct *demographic groups* $G_1, G_2, \ldots, G_k$. (These groups can overlap; for example G_1 can be equal to all residents of New York State, and G_2 can be equal to all people with a degree in computer science.) The site has contracts with m different *advertisers*, to show a certain number of copies of their ads to users of the site. Here's what the contract with the i^{th} advertiser looks like:

- For a subset $X_i \subseteq \{G_1, \ldots, G_k\}$ of the demographic groups, advertiser i wants its ads shown only to users who belong to at least one of the demographic groups in the set X_i.
- For a number r_i, advertiser i wants its ads shown to at least r_i users each minute.

Now, consider the problem of designing a good *advertising policy* — a way to show a single ad to each user of the site. Suppose at a given minute, there are n users visiting the site. Because we have registration information on each of these users, we know that user j (for $j = 1, 2, \ldots, n$)

belongs to a subset $U_j \subseteq \{G_1, \ldots, G_k\}$ of the demographic groups. The problem is: is there a way to show a single ad to each user so that the site's contracts with each of the m advertisers is satisfied for this minute? (That is, for each $i = 1, 2, \ldots, m$, at least r_i of the n users, each belonging to at least one demographic group in X_i, are shown an ad provided by advertiser i.)

Give an efficient algorithm to decide if this is possible, and if so, to actually choose an ad to show each user.

(3) Some of your friends on the West Coast have decided to carpool to work. None of them likes driving very much, so they want to make sure that any carpool arrangement they agree upon is fair, and doesn't overload any individual with too much driving. Some sort of simple round-robin scheme is out, because none of them goes to work every day, and so the subset of them in the car varies from day to day.

Here's one way to define *fairness*, based on relative allocation of "driving obligations." Let the people be labeled $S = \{p_1, \ldots, p_k\}$; suppose the carpool plan lasts for d days; and on the i^{th} day a subset $S_i \subseteq S$ of the people go to work. If p_j does not go to work on day i, then she does not incur any driving obligation. If she does go to work, then she incurs a driving obligation equal to the fraction of the car that she occupied: $1/|S_i|$, where $|S_i|$ is the number of people in the set S_i. Accumulating this over all the days, the total driving obligation Δ_j for p_j is defined to be

$$\Delta_j = \sum_{i \text{ so that } p_j \in S_i} \frac{1}{|S_i|}.$$

(Note that this is precisely the expected number of times that p_j would have driven, had a driver been chosen uniformly at random from among the people going to work each day.)

Ideally, we'd like to say that p_j should only have to drive at most Δ_j times; unfortunately, Δ_j may not be an integer, and so we may have to round up. Thus, let's say that a *driving schedule* is a choice of a driver for each day — i.e. a sequence $p_{i_1}, p_{i_2}, \ldots, p_{i_d}$ with $p_{i_t} \in S_t$ — and that a *fair driving schedule* is one in which each p_j is chosen as the driver on at most $\lceil \Delta_j \rceil$ days. (Here $\lceil \Delta_j \rceil$ denotes the smallest integer that is greater than or equal to Δ_j.)

Prove that for any sequence of sets $S_1, \ldots, S_d$, there exists a fair driving schedule, by giving an efficient algorithm to compute a fair driving schedule for any input instance.