The Design and Analysis of Algorithms	Problem Set 6
CS 681 Fall 2003	Due Friday, December 5, 2003

Solve 2 of the following 3 problems. You may solve all 3 for extra credit. We will maintain a FAQ for the problem set on the course Web page.

(1) (15 points) Consider the following problem. There is a set U of n nodes (users) (e.g., these are locations that need to access a service, such as a Web server). You would like to place servers at multiple locations. Suppose you are given a set S possible sites that would be willing to serve as locations for the servers. For each site $s \in S$ there is a fee $f_s \geq 0$ for placing a server at that location. Your goal will be to approximately minimize the cost while providing the service to each of the customers. So far this is very much like the set-cover problem, the places s are sets, their weight is f_s, and we want to select a subset that covers all users. There is one extra complication: users $u \in U$ can be served from multiple sites, but there is an associated cost d_{us} for serving user u from site s. When the value d_{us} is very high, we do not want to serve user u from site s; and in general the service cost d_{us} serves as an incentive to serve customers from "nearby" servers whenever possible.

So here is the algorithmic problem: Given the sets U, and S, and costs f and d, you need to select a subset $A \subseteq S$ to activate (at the cost of $\sum_{s \in A} f_s$), and assign each user u to the active server where it is cheapest to be served $\min_{s \in A} d_{us}$. The goal is to minimize the overall cost $\sum_{s \in A} f_s + \sum_{u \in U} \min_{s \in A} d_{us}$. Give an $H(n)$-approximation for this problem.

Hint: note that if all service costs d_{us} are 0 or infinity, than this problem is exactly the setcover problem: f_s is the cost of the set named s, and d_{us} is 0 if node u is in set s, and infinity otherwise.

(2) (15 points) In class we gave a 2-approximation for a load balancing problem on m machines. In this problem you will design an improved approximation algorithm in the special case of 2 machines. Recall that the input to the problem had n jobs and m machines, and a processing times p_{ij} that are the load that job j would present if assigned to machine i. The hart of the 2-approximation algorithm given in class was an algorithm for the following "decision" problem: given an extra parameter L either show that there is no assignment of jobs to machines with maximum machine load at most L or find an assignment with load at most $2L$.

(a.) Assume that we are given a target load L_i for all machines i, and a value L'. Assume that for all job j and machines i, the value p_{ij} is either infinite, or $\leq L'$. Give a polynomial time algorithm that,

- either shows that there is no assignment of jobs to machines where the load of machine i is at most L_i for all i, or
- finds an assignment where the load of machine i is at most $L_i + L'$ for all machines i.

(b.) Assume that the number of machines m is 2. Give an algorithm that, given an extra parameter L, either shows that there is no assignment of jobs to machines with maximum machine load at most L, or finds an assignment with load at most $1.5L$. (Note that this leads to a 3/2-approximation). Hint: use part (a) with $L' = L/2$.

(3) (15 points) In this problem, we will consider the following simple randomized vertex cover algorithm.

```
Start with S = ∅.
While S is not a vertex cover,
   Select an edge e not covered by S.
   Select one end of e at random (both end equally likely)
   Add selected node to S
Endwhile
```

We will be interested in the expected cost of a vertex cover selected by this algorithm.

(a.) Is this algorithm a c-approximation algorithm for the minimum weight vertex cover problem for some constant c? Prove your answer.

(b.) Is this algorithm a c-approximation algorithm for the minimum cardinality vertex cover problem for some constant c? Prove your answer.

Hint: For an edge, let p_e denote the probability that edge e is selected as a uncovered edge in this algorithm. Can you express the expected value of the solution in terms of these probabilities? To bound the value of an optimal solution in terms of the p_e probabilities, try to bound the sum of the probabilities for the edges adjacent to a given vertex v: $\sum_{e \text{adjacent to } v} p_e$.