CSE 521: Design and Analysis of Algorithms — Midterm Exam

Covers everything so far. — May 14th, 2009.

Due: IN CLASS, May 21st, 2009.

You are allowed to use notes that you have taken in class, notes and slides posted on the class web page, the Kleinberg-Tardos book, and the provided homework solutions, but nothing (and no one) else. There is no collaboration allowed on this exam.

You should do any **4** out of the following **5** problems. All problems are worth the same number of points.

Problems

1. Given a graph $G = (V, E)$, a *vertex cover* of G is a set of vertices $C \subseteq V$ such that each edge of G has at least one endpoint in C.

   (a) Consider the following algorithm for vertex cover.

      i. Start with $C \leftarrow \emptyset$.

      ii. Pick an edge $e = \{u, v\}$ such that $\{u, v\} \cap C = \emptyset$, and add an arbitrary endpoint of e to C.

      iii. If C is a vertex cover, halt; else go to step (ii).

   Give an instance on which this algorithm may return a set which is $\Omega(n)$ times bigger than the optimal (smallest) vertex cover.

   (b) Now suppose we randomize the algorithm: In step (ii), we throw a random endpoint of $e = \{u, v\}$ into C. If k is the size of an optimal vertex cover of G, show that $\mathbb{E}[|C|] \leq 2k$, i.e. this is a 2-approximation algorithm.

2. Alice wants to throw a party and is deciding who to call. She has n people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick as many people as possible, subject to two constraints: At the party, each person should have at least five other people they know, *and* five other people that they don't know.

   Give an efficient algorithm that takes as input a list of n people and the list of pairs who know each other and outputs the best choice of party invitees.

3. Let $G = (V, E)$ be a directed graph, with source $s \in V$ and sink $t \in V$, and non-negative edge capacities. Give a polynomial-time algorithm to decide whether G has a *unique* minimum s-t cut, i.e. one whose capacity is *strictly less* than *every other* s-t cut.

   Hint: Consider the residual graph at the termination of the Ford-Fulkerson algorithm, and define minimum s-t cuts based on this graph in two natural ways. When are these two cuts actually the same cut?

4. A subset of the nodes of a graph G is a *dominating set* if every node of G is adjacent to some node in the subset. Let

$$\text{DOMINATING-SET} = \{\langle G, k\rangle : G \text{ has a dominating set with } k \text{ nodes}\}.$$

Show that DOMINATING-SET is NP-complete by giving a reduction from VERTEX-COVER.

5. In a satisfiable system of linear inequalities

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &\leq b_1 \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n &\leq b_n \end{aligned}$$

we describe the jth inequality as *forced-equal* if it is satisfied with equality by *every* solution $x = (x_1, \ldots, x_n)$ of the system. Equivalently, $\sum_i a_{ji}x_i \leq b_j$ is *not* forced-equal if there exists an $x = (x_1, \ldots, x_n)$ that satisfies the whole system and such that $\sum_i a_{ji}x_i < b_j$.

For example, in

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ -x_1 - x_2 &\leq -2 \\ x_1 &\leq 1 \\ -x_2 &\leq 0 \end{aligned}$$

the first two inequalities are forced-equal, while the third and fourth are not. A solution x to the system is called *characteristic* if, for every inequality I that is not forced-equal, x satisfies I without equality. In the instance above, such a solution is $(x_1, x_2) = (-1, 3)$, for which $x_1 < 1$ and $-x_2 < 0$ while $x_1 + x_2 = 2$ and $-x_1 - x_2 = 2$.

(a) Show that any satisfiable system has a characteristic solution.

(b) Given a satisfiable system of linear inequalities, show how to use linear programming to determine which inequalities are forced-equal, and to find a characteristic solution.