CSc 573 **Final Exam** **Spring 2007**

Name ____

Instructions This exam consists of three problems worth a total of 100 points.

Choose *one* of Problems (1) or (2). You must do Problems (3) and (4). In other words, you will do a total of three questions. If you do more than three questions, only three will be graded.

You have 2 hours. The exam is closed book, closed notes, and closed calculator. Good luck!

(1) **(Steiner trees)** (40 points) Given an undirected graph $G = (V, E)$ and a vertex subset $L \subseteq V$, a *Steiner tree* T on vertices L is an edge subset $T \subseteq E$ such that T is a tree that contains all vertices in L. In other words, a Steiner tree on L must contain the vertices in L, but may also contain other vertices outside L in order to connect the vertices in L through edges in G.

Let

$$\text{SteinerTree} := \left\{ \langle G, L, k \rangle \ : \ \begin{pmatrix} G \text{ is an undirected graph that contains} \\ \text{a Steiner tree } T \text{ on } L \text{ with } |T| \leq k \end{pmatrix} \right\}.$$

In other words, the language SteinerTree consists of instances that have a Steiner tree with at most k edges.

Prove that SteinerTree is NP-complete.

(Hint: Use a reduction from the Vertex Cover Problem, which is to decide the language

$$\text{VertexCover} := \left\{ \langle G, k \rangle \ : \ \begin{pmatrix} G \text{ is an undirected graph that contains} \\ \text{a vertex subset } C \text{ such that every edge} \\ \text{in } G \text{ has at least one endpoint in } C \end{pmatrix} \right\}.$$

(This page is blank.)

(2) **(Multiprocessor scheduling)** (40 points) Given tasks $1, 2, \ldots, n$ that have associated integer running times $t_1, t_2, \ldots, t_n$, a collection of $k \geq 2$ processors, and a time bound T, the Multiprocessor Scheduling Problem is the following. Find an assignment of the n tasks to the k processors, represented by a partition of $\{1, \ldots, n\}$ into k sets $P_1, P_2, \ldots, P_k$, such that the completion time of the tasks under the assignment is at most T, or more formally:

$$\max_{1 \leq i \leq k} \left\{ \sum_{j \in P_i} t_j \right\} \leq T.$$

Let

$$\text{ProcSched} := \left\{ \langle n, t_1, t_2, \ldots, t_n, k, T \rangle : \begin{pmatrix} \text{The } n \text{ tasks with integer running times} \\ t_1, \ldots, t_n \text{ can be assigned to } k \text{ processors} \\ \text{with completion time at most } T \end{pmatrix} \right\}.$$

Prove that ProcSched is NP-complete.

(Hint: Use a reduction from the Sum Split Problem, which is to decide the language

$$\text{SumSplit} := \left\{ \langle S \rangle : \begin{pmatrix} S \text{ is a multiset of integers that has a partition} \\ \text{into two sets } A, B \text{ such that } \sum_{a \in A} a = \sum_{b \in B} b \end{pmatrix} \right\}.$$

(This page is blank.)

(3) **(Containment of DFAs)** (30 points) Let

$$\text{DFAContain} := \Big\{ \langle M_1, M_2 \rangle \;:\; M_1, M_2 \text{ are DFAs with } L(M_1) \subseteq L(M_2) \Big\}.$$

Prove that DFAContain is a member of $\mathcal{NL}$.

(This page is blank.)

(4) **(Directed Get Home)** (30 points) The *Directed Get Home Game* is the same as the undirected Get Home Game, except it is played on *directed* graph. More precisely, Directed Get Home is the following two-player game.

Directed Get Home is played by players I and II on a directed graph $G = (V, E)$, with three special vertices $s, x, y \in V$ that are all distinct. Vertex s is the *start vertex*, vertex x is the *home vertex* for player I, and vertex y is the *home vertex* for player II.

Play starts from node s with player I moving first. Moves alternate between players I and II. Each move extends a directed path P from s by choosing an edge $e \in E$ that leaves the last vertex on P. The chosen edge e must *not* go to a vertex already visited by P.

Player I wins if the path reaches his home vertex x, or if player II has the move and cannot extend the path. Player II wins if the path reaches his home vertex y, or if player I has the move and cannot extend the path.

Let

$$\text{DirGetHome} := \left\{ \langle G, s, x, y \rangle \;:\; \begin{pmatrix} \text{Player I has a winning strategy in the Directed} \\ \text{Get Home Game played on directed graph } G \\ \text{with start vertex } s \text{ and home vertices } x, y \end{pmatrix} \right\}.$$

Prove that DirGetHome is PSpace-complete.

(This page is blank.)