CPSC 455/555 // ECON 425/563, Fall 2011, Exam 1

Answer all of the questions. Please remember to write your name, the course number, and today's date on all blue books that you submit.

This is a closed-book exam; please do not refer to any books or notes, and please do not talk to any of the other students.

Question 1:

(a) (15 points) Recall the local connection game of Section 19.2 and HW1. Prove that, for $\alpha > n^2$, where α is the edge cost, and n is the number of nodes, all Nash equilibria are trees.

(b) (5 points) Figure 1 contains an instance of the global connection game of Section 19.3 and HW1. What are the Price of Anarchy and the Price of Stability in this instance?

Question 2:

(a) (10 points) In the AS graph in Figure 2, d is the destination AS, the ASes 1 through $n+2$ are sources, and $n \geq 2$. Directed edges point from customers to providers. All source nodes export their preferred routes to all of their neighbors. AS $n+2$ assigns value n^2 to route $n+2--n+1--1--d$ and value 1 to all other routes. AS $n+1$ assigns value i to route $n+1--i--d$, for $1 \leq i \leq n$. For $1 \leq i \leq n$, AS i assigns value 1 to all customer routes and value 0 to all other routes. Does BGP converge to a stable routing tree on this interdomain-routing instance? If so, does it converge to an optimal routing tree? Briefly justify your answers.

(b) (10 points) In the AS graph in Figure 3, directed edges point from customers to providers, and undirected edges are peer edges. The destination is d, and all other ASes are sources.

Suppose that the route-export policies of source ASes 1, 2, 3, and 4 satisfy the Gao-Rexford scoping constraints. Suppose further that, under normal operating conditions, AS 2 uses the route $2--d$, and AS 4 uses the route $4--3--d$. Why would AS 2 be able to switch to $2--1--d$ if the edge $2--d$ went down but AS 4 not be able to switch to $4--1--d$ if the edge $4--3$ went down?

Question 3:

Recall that the algorithmic mechanism-design problem of *task allocation* is defined as follows. There are n agents and k tasks $\{z_1, \ldots, z_k\}$. Agent i's type is $T^i = (t_1^i, \ldots, t_k^i)$, where t_j^i is the minimum time in which agent i could execute task z_j, for $1 \leq j \leq k$. The feasible outcomes, also known as feasible allocations, are all of the partitions of the task set Z into n subsets $Z = Z^1 \sqcup \cdots \sqcup Z^n$, where Z^i is the set of tasks assigned to agent i. So each task z_j is assigned to exactly one agent. The valuation that agent i of type T^i assigns to an allocation $Z = Z^1 \sqcup \cdots \sqcup Z^n$ is $v^i(T^i, Z) = -\sum_{z_j \in Z^i} t_j^i$. This can be interpreted to mean that agent i wants to spend as little time as possible on the (sequential) execution of the tasks assigned to him. If Z is an allocation, then the *makespan* of Z is the maximum total time spent by any agent on the execution of his assigned tasks, *i.e.*, $Makespan(Z) = \max_i \sum_{z_j \in Z^i} t_j^i$. An *optimal allocation* is one that achieves the smallest possible makespan.

In the *MinWork* mechanism, agents are asked simply to report their types. If MinWork receives inputs $A^i = (a_1^i, \ldots, a_k^i)$, for $1 \leq i \leq n$, it assigns task z_j to an agent i who reports the smallest time a_j^i, and it pays agent i the second smallest reported time for each task assigned to him, i.e.,

$$p^i(A^1, \ldots, A^n) = \sum_{z_j \in Z^i} \min_{i' \neq i} a_j^{i'}.$$

(a) (7 points) Prove that MinWork is a polynomial-time computable, strategyproof mechanism.
(b) (8 points) Prove that MinWork produces an n-approximately optimal allocation, *i.e.*, that the makespan of any allocation produced by MinWork is at most n times the makespan of an optimal allocation.
(c) (5 points) Why do we not expect to be able to design a polynomial-time computable, strategyproof mechanism that computes an optimal allocation?

Question 4:
Consider a combinatorial auction of the item set $S = \{s_1, \ldots, s_m\}$. Recall that a bidder's valuation function $v()$ is called *symmetric* if the value that this bidder assigns to a bundle depends only on the bundle's size, *i.e.*, $v(S_1) = v(S_2)$ for all S_1 and S_2 with $|S_1| = |S_2|$. Recall further that a symmetric valuation function is called *downward sloping* if, for any bundle T, there is a decreasing sequence of prices $p_1 \geq p_2 \geq \cdots \geq p_{|T|} \geq 0$ such that $v(T) = \sum_{1 \leq j \leq |T|} p_j$.
(a) (10 points) Prove that any downward-sloping, symmetric valuation function can be represented by an OR-of-XOR bid of size m^2.
(b) (10 points) Prove that, if the sequence of prices $p_1 > p_2 > \cdots > p_m > 0$ is strictly decreasing and greater than zero, then the downward-sloping, symmetric valuation function cannot be represented by an XOR bid of size less than $2^m - 1$.

Question 5:
(a) (7 points) Give the definition of *ex-post Nash equilibrium* and explain how it differs from a dominant-strategy equilibrium.
(b) (7 points) Give the definition of *locally envy-free equilibrium* as the term is used in sponsored-search auctions.
(c) (6 points) Give the definition of *mixed-strategy Nash equilibrium* and an example of a game that does not have a pure-strategy Nash equilibrium but does have a mixed-strategy Nash equilibrium.

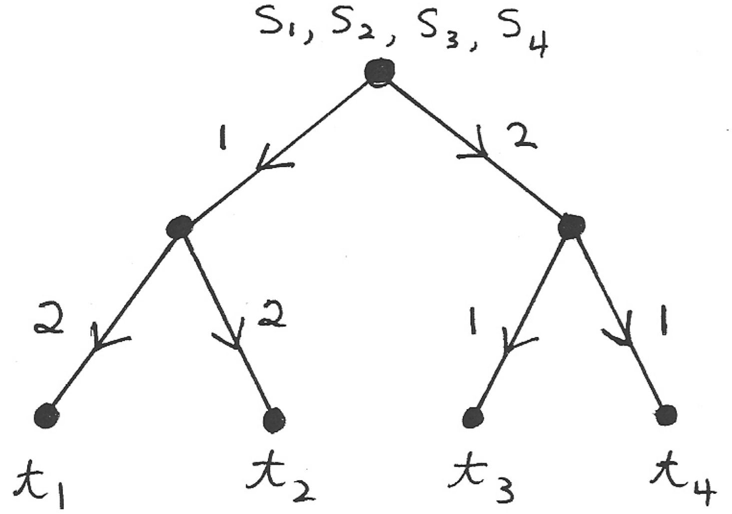

Figure 1

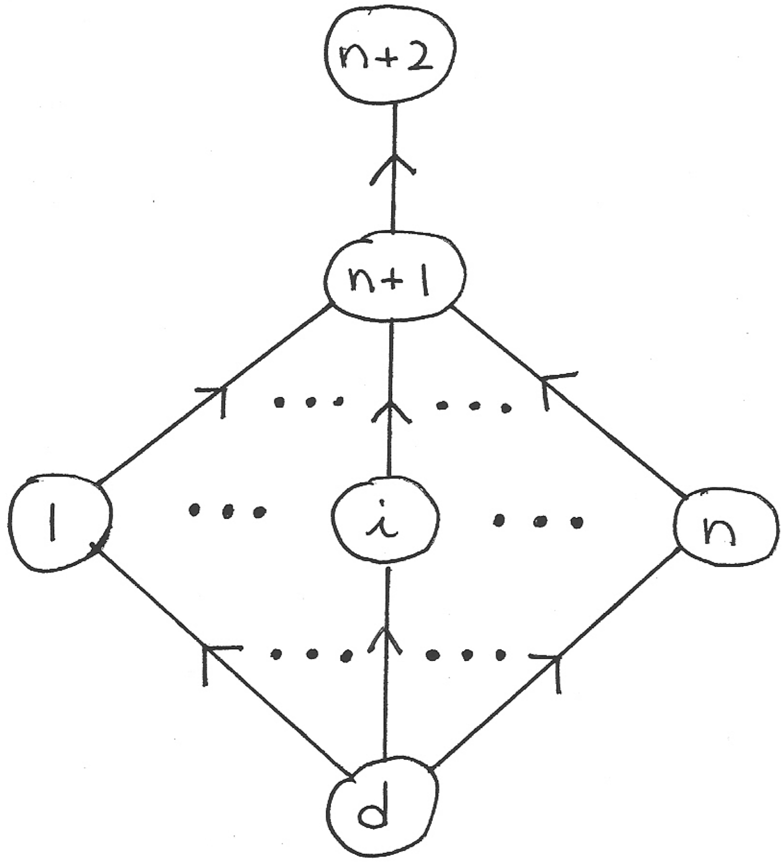

Figure 2

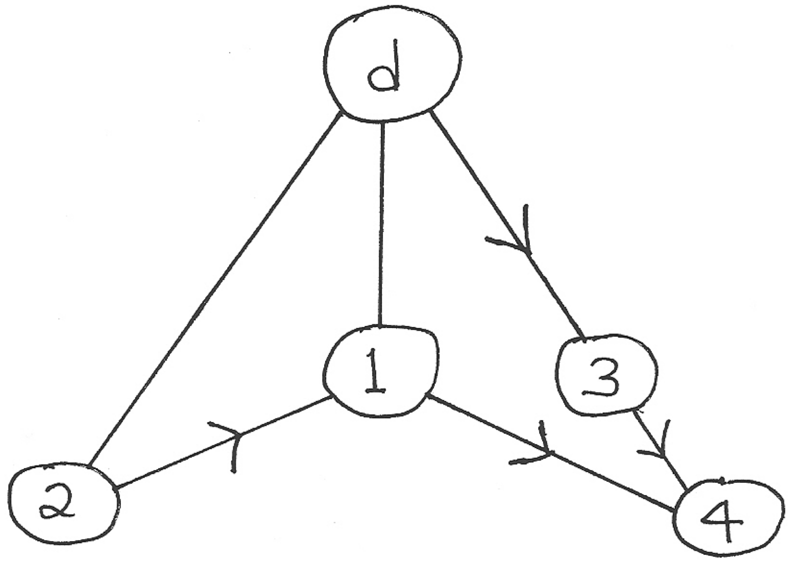

Figure 3