CS 580: Algorithmic Game Theory, Fall 2025
Practice Exam

Instructions:

1. We advise you to read all the instructions and problems carefully before start writing the solutions.
2. There are five problems in total, each of 20 points. That is 100 points in total.
3. First problem is compulsory, while you are asked to do any 4 out of the next 5.
4. Even if you are not able to solve a problem completely, do submit whatever you have. Partial proofs, high-level ideas, examples, and so on.
5. Be precise and succinct in your argument.

---

1. Answer the following. Each question is worth 5 points.

   (a) Give an example of a fair-division instance with indivisible items and an EFX allocation for the instance that is not PO.

   (b) In a stable roommate problem there is a set of m dorm rooms each of which can accommodate exactly two students, and a set n of students where $m > n/2$. Each student has a total ordering on all the other students (non-bipartite), and prefers to have a roommate then to live alone. Given an assignment of students to rooms, a pair of students (s_1, s_2) is unstable if they both prefer each other more than their current assignment. An assignment is called stable if there is no unstable pair. Construct an example that has no stable assignment.

   Hint: *Think of a three node graph.*

   (c) Consider the "Knapsack Auction," where there is a knapsack with total weight W, and n bidders each with a public weight w_i and private valuation v_i, for $i = 1, 2, \ldots, n$. The mechanism wishes to select a subset of agents with total weight at most W. A selected agent gets utility v_i, and an unselected agent, 0. Consider the greedy allocation mechanism which collects bids from the agents (reported valuations), sorts them in decreasing order of bid-per-weight $\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \cdots \geq \frac{b_n}{w_n}$, then allocates the players one by one until there is no room left.

   If there are *two* knapsacks, with weights W_1 and W_2, and the mechanism first allocates greedily up to W_1 in the first bag, then, with the remaining jobs, greedily up to W_2 in the second.

   Is this a monotone allocation rule? If yes, prove it. If no, give a counterexample.

   (d) Give an example of a two-player game with pure Nash equilibria, where best response dynamics can cycle.

Do any four out of the following five problems.

2. Consider a game where n players are allocating a shared bandwidth of 1. Each player i chooses an amount $1 \geq x_i \geq 0$ and the utility of player i is $U_i(x_i, \boldsymbol{x}_{-i}) = x_i \cdot (1 - \sum_{j=1}^{n} x_j)$. Note that if the sum of x_j's is greater than 1, then all players have a negative utility.

   (a) Give a Nash equilibrium of this game.

   (b) Is it a potential game? Prove your answer.

3. Consider a variant of the knapsack auction in which both the valuation v_i and the size w_i of each bidder i are private. A mechanism now receives both bids $\boldsymbol{b} = (b_1, \ldots, b_n)$ and reported sizes $\boldsymbol{a} = (a_1, \ldots, a_n)$ from the bidders, where n is the number of bidders. Assume each v_i, w_i, b_i, a_i to be positive. An allocation rule $x(\boldsymbol{b}, \boldsymbol{a})$ now specifies the amount of capacity allocated to each bidder, as a function of the bids as well as the reported sizes. Feasibility of an allocation requires that the total capacity allocated should not exceed the total capacity W of the shared resource, i.e., $\sum_{i=1}^{n} x_i(\boldsymbol{b}, \boldsymbol{a}) \leq W$ for every $\boldsymbol{b}$ and $\boldsymbol{a}$. We define the utility of a bidder i as $v_i - p_i(\boldsymbol{b}, \boldsymbol{a})$ if she gets her desired capacity (i.e., $x_i(\boldsymbol{b}, \boldsymbol{a}) \geq w_i$) and as $-p_i(\boldsymbol{b}, \boldsymbol{a})$ otherwise. Note that this is not a single-parameter environment.

   Consider the following mechanism. Given bids $\boldsymbol{b}$ and reported sizes $\boldsymbol{a}$, the mechanism runs the greedy knapsack auction: It first sorts the bidders in the decreasing order of $\frac{b_i}{a_i}$. Then moving from left to right in the sorted list, when it is bidder i's turn it awards her a capacity equal to her reported size a_i if a_i amount of space is available in the sack ($x_i(\boldsymbol{b}, \boldsymbol{a}) = a_i$) and subtracts a_i from the sack space, otherwise i gets no allocation ($x_i(\boldsymbol{b}, \boldsymbol{a}) = 0$). And it continues allocating this way until the end of the list is reached. Every *winning bidder* i (i.e., one which does get allocated its a_i) is charged a price of p_i defined as "the minimum bid which allows i to be a winning bidder, by fixing all the bids b_{-i}, reported sizes a_{-i} and its own a_i". A losing bidder neither gets anything, nor pays anything.

   A truthful strategy for a player i in this case is one, where $b_i = v_i$, as well as $a_i = w_i$. Prove that this is a DSIC mechanism.

   [Hint: You may consider the cases of a buyer being winning or losing when she bids truthfully, and analyze the dominance of the truthful strategy in each case separately]

4. Consider a load balancing game with n jobs and a set M of machines. Each job is a player who chooses a machine to run on, and tries to minimize its *completion time*. Job j has size p_j, and any jobs can choose any of the m machines. A machine releases a job only after finishing all of its jobs, i.e., if $s = (s_1, \ldots, s_n) \in [m]^n$ is the strategy profile, then the completion time for all the jobs assigned to machine $m \in M$ is $\sum_{i \in [n]: s_i = m} p_i$.

   Suppose the social welfare is given by the maximum completion time. That is, $SW(s) = \max_{m \in M} \sum_{i \in [n]: s_i = m} p_i$. Show that the Price of Anarchy is upper-bounded by 2.

5. Consider the following weighted generalization of the network cost-sharing game. For each player i, we have a weight $w_i > 0$. As before, each player selects a single path connecting her source and sink. But instead of sharing edge cost equally, players are now assigned cost shares in proportion to their weight. In particular, for a strategy vector S and edge e, let S_e

denote those players whose path contains e, and let $W_e = \sum_{i \in S_e} w_i$ be the total weight of these players. Then player i pays $c_e \cdot \frac{w_i}{W_e}$ for each edge $e \in P_i$. Note that if all players have the same weight, this is the original game.

This game does not have an *exact* potential function.

(a) Consider the potential-like function $\Phi(P) = \sum_{e:e \in P}(c_e/W_e)$, where W_e is the total weight of players on e. Show that function Φ satisfies,

$$\forall i, \forall \text{ paths } P_i, P_i' : \quad \Big(\Phi(P_i, P_{-i}) - \Phi(P_i', P_{-i})\Big) \cdot \Big(C_i(P_i, P_{-i}) - C_i(P_i', P_{-i})\Big) \geq 0$$

$$\forall i, \forall \text{ paths } P_i, P_i' : \quad \Phi(P_i, P_{-i}) = \Phi(P_i', P_{-i}) \quad \Longleftrightarrow \quad C_i(P_i, P_{-i}) = C_i(P_i', P_{-i})$$

(b) Using the above, show that the game has a pure Nash equilibrium.

6. Fair division of chores is the problem of allocating a set of M *chores* among n agents who experience disutility in doing these chores. Agent $i \in [n]$ has dis-utility d_{ij} for doing chore $j \in M$, and her disutility for doing a set $S \subseteq M$ of chores is additive, *i.e.,* $\sum_{j \in S} d_{ij}$. An allocation $(A_1, \ldots, A_n)$ is EF1 if no agent envies another agent up to the removal of one chore from her own bundle. Formally,

$$\forall k \in [n], \quad \sum_{j \in A_i} d_{ij} - \max_{j' \in A_i} d_{ij'} \leq \sum_{j \in A_k} d_{ij}.$$

Design an algorithm to find an EF1 allocation.

[Hint: Think round-robin]