CSCI 357: HW 1 (due 02/18/2025) **Your Name:** ____________________

Guideline: These are short-answer questions that must be done individually. The goal is to practice the definitions from lecture. Write your answer in the space provided. No justification is required. The completed HW must be handed back in class.

Problem 1. (Forming a Game) Suppose you have to prepare a piece to perform at your music club. You only care about impressing a friend, who may or may not attend. You must decide whether to practice the piece (which is costly), or just "wing it" which is easy but may lead to embarrassment if your friend attends.

In particular, if you practice the piece and your friend attends the event, you are happy (utility 5), while if your friend does not attend, you feel like your effort was wasted (utility -5). If you do not practice, and your friend attends, you feel embarrassed (utility -10). If you do not practice and your friend skips, you don't care (utility 0).

On your friend's side, if they show up and you perform a well-practiced piece, they enjoy it (utility 10), otherwise, they feel like they wasted their time (utility -5). If your friend does not attend the event, their utility is 0 regardless of how you perform.

(a) Formalize the normal-form game by drawing the payoff matrix in the space provided.

(b) One of the players has a dominant-strategy. True or False?

(c) State all pure-strategy Nash equilibria (NE) of the game. Your NE must be an **action profile**.

(d) Is there a Nash equilibria from part (c) which is Pareto optimal?

Problem 2. (Machine Scheduling: Parkes & Seuken) In a load balancing games, we have jobs that need to get executed and machines that can process jobs. The jobs here are players and can choose which machine to run on. Consider such a game with two identical machines and four jobs. Each job i has size $s_i > 0$, representing the length of the task. The sizes are $s_1 = s_2 = 2$ and $s_3 = s_4 = 1$, so that there are two large tasks and two small tasks. Each machine's speed is 1 unit per second. The completion time of a job i assigned to machine j is equal to the time when machine j completes all jobs assigned to it (that is, all jobs assigned to a particular machine finish at the same time)

Each job selfishly selects a machine to minimize its completion time.

(a) Identify and state two pure-strategy Nash equilibria of this game.

(b) What is a socially optimal assignment; that is, the assignment of jobs to machine that minimizes the maximum completion time across both machines (this is termed the "makespan").

Iterated Elimination of Dominated Strategies

Problem 3. In class we discussed iterated elimination of *weakly* dominated strategies. We can also consider iterated elimination of *strictly* dominated strategies. An action $a_i \in A_i$ is **strictly dominated** if $u_i(a_i, a_{-i}) < u_i(a'_i, a_{-i})$ for some $a'_i \in A$ and for all $a_{-i} \in A_{-i}$.

Algorithm 1 describes the method where the input is a set of n action sets $A_1, \dots, A_n$ and a set of a n utility functions $u_1, \dots, u_n$, where each $u_i : A_1 \times \cdots A_n \to \mathbb{R}$

(a) Which strategy profiles survive the iterated elimination of strictly dominated strategies in the following normal-form game?

	L	C	M
T	$0, 6$	$3, 4$	$1, 5$
M	$2, 1$	$3, 3$	$2, 4$
B	$2, 3$	$5, 1$	$3, 4$

Algorithm 1: Iterated Elimination of Strictly Dominated Strategies

Input: $A_1, \ldots, A_n, u_1, \ldots, u_n$

Let S_i denote the set of undominated actions for agents i
For each player i, initialize $S_i = A_i$
Initialize DominatedActionFound = True
while DominatedActionFound **do**
    if *there exists some agent* i*, some action* $a_i \in S_i$ *and some action* $a'_i \in S_i$ *such that* $u_i(a_i, a_{-i}) < u_i(a'_i, a_{-i})$ *for all* $a_{-i} \in S_{-i}$ **then**
        Remove action a_i from S_i
    else
        DominatedActionFound = False
    end
end
return $S_1, \ldots, S_n$

(b) Give an example of a game with 2 players and 2 actions where **no action** can be eliminated.

(c) Let m be the maximum number of actions available to any player, and n, the number of players. What is the worst-case running time of iterated elimination of strictly dominated strategies? Give the best possible Big Oh bound in terms of n and m. Assume that the utility comparison of two distinct action profiles (a_i, a_{-i}) and (a'_i, a_{-i}) can be done in constant time.