Due date: 11:59 pm on Wednesday, February 1, 2023.

Notion: 4 Late days were user for this assignment. Sorry for the late submission since I had a fever these days and felt very weak. I think I will submit the remaining homework on time.

Logistical questions:

1. Did you collaborate with anyone? **No**
2. How long did this assignment take you? About **24** hours
3. Do you have any comments on the assignment? First of all, I think the homework is very excellent since it can deep my understanding of what professor taught in lecture. But one problem is that when I first look through the problems, I feel that I have no idea about most of them. I think that's because the content taught in the lecture is too conceptual without enough corresponding examples, and the homework is more like application, which need us to "jump our mind". I think more examples in the lecture will help a lot.

Problem 1: Iterated Elimination [20 pts]

(a) (10 pts) Prove that iterated elimination of strictly dominated actions never removes an action in the support of *any* mixed-strategy Nash equilibrium.

a) Proof by contradiction:

Assume agent i, action profile $(a_1, \cdots a_n) = A_i$, action a_k is strictly dominated by action a_m, which indicates that $u_i(a_m, j) \geq u_i(a_k, j)$, where $j \subseteq A_{-i}$ and $u_i(a_m, j) > u_i(a_k, j)$ for some $j \subseteq A_{-i}$, so a_k is removed

Now ~~suppor~~ assume action a_k in the support of MSNE s^*, then we have

$$u_i(j, s^*_{-i}) \geq u_i(j', s^*_{-i})$$

for all $j \in \sigma(s^*_i)$, all $j' \in A_i$, and all agents i.

But a_k is strictly dominated by action a_m, so we can construct j' such that changing probability $s_{ik} = 0$, $s_{im} = s_{im} + s_{ik}$.

In this case $\Rightarrow u_i(j, s^*_{-i}) < u_i(j', s^*_{-i})$, Contradiction!

Therefore, the statement is proved to be True.

Solution. □

(b) (10 pts) What is the time complexity of iterated elimination of strictly dominated actions in a game with n players, each with m actions? Show your work and explain your reasoning.

Solution. For time complexity, we consider the worst case, that is, every time we remove only one action. For every remove, we check every action file, denote the time complexity as T. We can calculate that T $= m^n + (m-1)^n + ... + 2^n$ □

Problem 2: Equilibria [20 pts]

		Player 2			Player 2	
		L	R		L	R
Player 1	T	(5, 5, 5)	(2, 6, 2)	T	(2, 2, 6)	(-1, 3, 3)
	B	(6, 2, 2)	(3,3,-1)	B	(3, -1, 3)	(0, 0, 0)
		N			F	

Table 1: Three-player normal-form game. Player 3 plays N or F, player 2 plays L or R, and player 1 plays T or B.

(a) (10 pts) Consider the three-player normal-form game in Table 1. Each player has two actions: (T, B) for player 1, (L, R) for player 2, and (N, F) for player 3. Player 3 gets to select the left or right payoff matrix, player 2 gets to select the column, and player 1 gets to select the row. For example, if they play (T, L, F), the payoffs are 2, 2, and 6 to players 1, 2, and 3 respectively. Find all pure-strategy Nash equilibria in this game, and list all pure-strategy Pareto-optimal outcomes of the game.

Solution. For this game, first look at Player 1: **T** is strictly dominated by **B**; then given that Player 1 will only conceivably play B, Player 2 can see that **R** dominates **L**; Lastly, given B and R, Player 3 will choose **F** obviously. Therefore, only one **PSNE: (B, R, F)**. By definition, **Pareto-optimal** outcomes are: (T, L, N), (T, R, N), (T, L, F), (B, L, N).

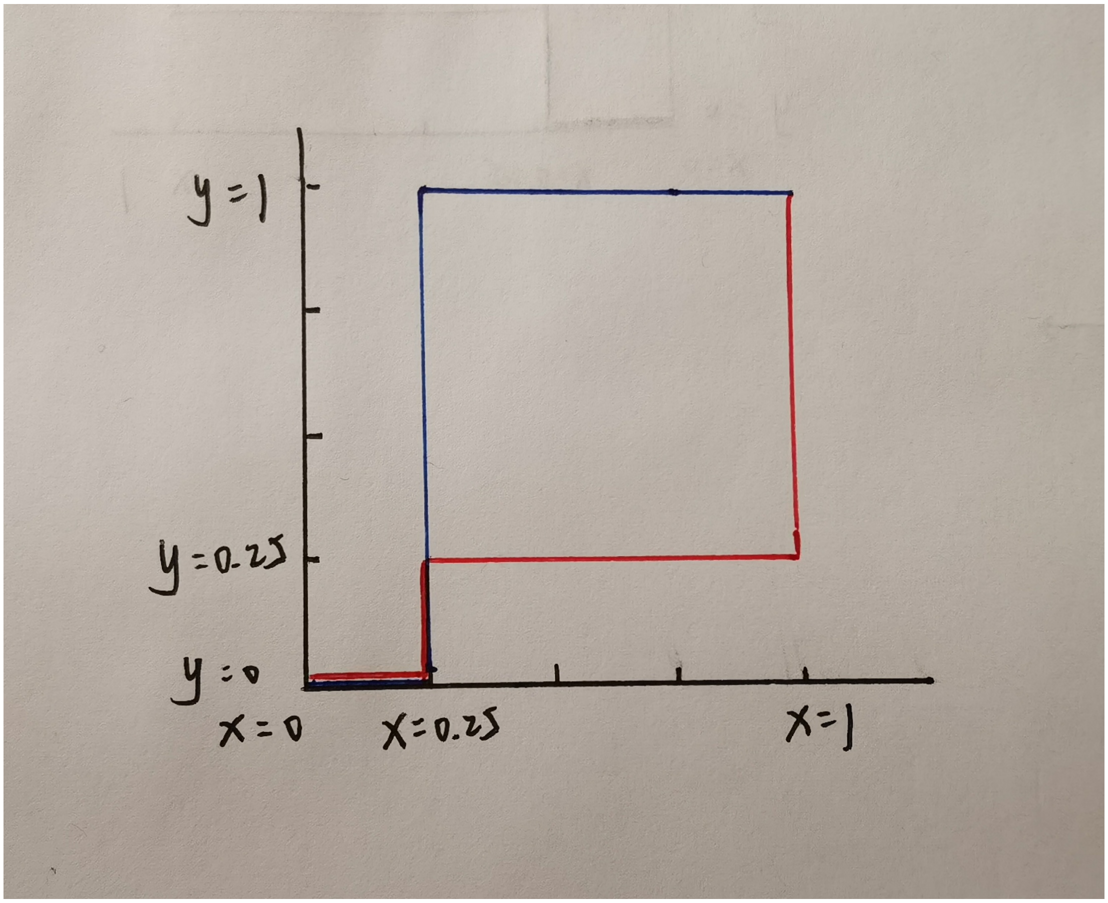
□

		Player 2	
		Stag	Hare
Player 1	Stag	400, 400	0, 100
	Hare	100, 0	100, 100

Table 2: Stag payoff matrix.

(b) (10 pts) In the game of Stag (Table 2), two hunters can decide whether to hunt for a stag (big prey) or a hare (small prey). The stag is hard to catch and they both need to agree, whereas the hare is easier to catch alone but less valuable. Plot the best responses and identify all the mixed and pure Nash equilibria of the game.

Solution. Let x denote the probability with which Player 1 plays Stag and let y denote the probability with which Player 2 plays Stag. For Player 1, utility = $400xy + 100(1-x)y + 100(1-x)(1-y) = 100x(4y-1) + 100$; For Player 2, utility = $400xy + 100x(1-y) + 100(1-x)(1-y) = 100y(4x-1) + 100$, where $0 <= x, y <= 1$. Therefore, Plot is shown below. **MSNE**: s1 = (0.25, 0.75), s2 = (0.25, 0.75); **PSNE**: (Stag, Stag), (Hare, Hare).



□

Problem 3: Correlated Equilibria [20 pts]

		Player 2 W	Player 2 G
Player 1	W	0, 0	0, 2
	G	2, 0	-4, -4

Table 3: Chicken payoff matrix.

Recall the game of Chicken, whose payoff matrix is reproduced in Table 3. Consider strategies in Chicken that take the following form, for some $x \in [0, 1]$:

$$\begin{pmatrix} 0 & x \\ 1-x & 0 \end{pmatrix}.$$

(a) (10 pts) Are these strategies correlated equilibria? Justify your answer.

Solution. Yes, these strategies are all correlated equilibra. We can easily find out that the **PSNE** for the problem is (W, G) and (G, W). Also, no matter what's the value of x, these strategies are also MSNE. Therefore, we can conclude that these strategies are all in Nash equilibrium. Since every Nash equilibrium is correlated equilibrium, these strategies are all correlated equilibrium for sure. □

(b) (10 pts) Are these strategies Pareto-optimal? Justify your answer.

Solution. Yes. First, without considering the randomization, we can easily know that (W, G) and (G, W) are Pareto-optimal. Consider the randomization, no matter what the value of x, the total utility is $2x + 2(1 - x) = 2$, which is a constant. Therefore, we can conclude that all these strategies Pareto-optimal. □

Problem 4: Auction Games [20 pts]

(a) (10 pts) Consider a first-price auction game with two agents and a single item to allocate. Agent 1 has a value of \$1 for the item and Agent 2 has a value of \$2 for the item. Agent 1 can bid $x \in [0, 10]$ and agent 2 can bid $y \in [0, 10]$. The agent with the higher bid wins the item and pays its bid amount. Ties are broken in favor of agent 2. If an agent is allocated the item, her utility is its value for the item minus its payment. Else, her utility is 0. Find all pure-strategy Nash equilibrium in this game and prove that your list of equilibria is exhaustive.

Solution. **PSNE**: (0, 0), (1, 1), (2, 2) I prove it by writing a python program to find the PSNE by definition.

```python
Agent_1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Agent_2 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

re = []
Agent_1_dic = {}
Agent_2_dic = {}

for i in Agent_1:
    for j in Agent_2:
        if i > j:
            u_A = 1-i
            u_B = 0
        else:
            u_A = 0
            u_B = 2-j
        Agent_1_dic[(i, j)] = u_A
        Agent_2_dic[(i, j)] = u_B

def judge_Agent_1(Tuple):
    for i in Agent_1_dic:
        if i[1] == Tuple[1]:
            if Agent_1_dic[i] > Agent_1_dic[Tuple]:
                return False
    return True

def judge_Agent_2(Tuple):
    for i in Agent_2_dic:
        if i[0] == Tuple[0]:
            if Agent_2_dic[i] > Agent_2_dic[Tuple]:
                return False
    return True

for i in Agent_1:
    for j in Agent_2:
        if judge_Agent_1((i, j)) & judge_Agent_2((i, j)):
            re.append((i, j))

for i in re:
    print(i)
```

```
D:\Anaconda\python.exe H:\CSC289\HW1_Q4.py
(0, 0)
(1, 1)
(2, 2)

Process finished with exit code 0
```

□

(b) (10 pts) Now, suppose there are two items A and B. Agent 1 needs both items: she values either item individually at \$0 but both items together at \$4. Agent 2 only needs one of the two items: she has value \$3 for either A or B, but also values both items together at \$3. Agent 1 can bid the $x \geq 0$ on item A and the same amount x on item B (she cannot bid on just one). Agent B can bid $y \geq 0$ on either item A or B, and must pick which (she cannot bid on both). Each item is assigned to the agent with the highest bid at a price equal to the bid on the item, and ties are broken in favor of agent 1. If an agent is allocated a collection of items, her utility is the value she has for her collection minus her payment. Else, her utility is 0. Are there any pure-strategy Nash equilibria in this game? Justify your answer.

Solution. Yes. Two PSNE: (2, 2(Player2 choose item A)), (2, 2(Player2 choose item B)). By analysis, we can easily get that Player 2 cannot get both item A and B, which means Player 2 only has three situations: get A, get B, or none. According to the given information, we can modify the above code a little to solve the problem, as shown below.

```
for i in Agent_1:
    for j in Agent_2:
        if i >= j:
            u_A = 4-2*i
            u_B = 0
        else:
            u_A = -i
            u_B = 3-j
        Agent_1_dic[(i, j)] = u_A
        Agent_2_dic[(i, j)] = u_B
```

```
D:\Anaconda\python.exe H:\CSC289\Q4_2.py
(2, 2)

Process finished with exit code 0
```

□

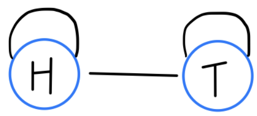
Problem 5: Succinct Games [20 pts]

		Player 2: H	Player 2: T
Player 1	H	1, -1	-1, 1
	T	-1, 1	1, -1

Table 4: Matching Pennies payoff matrix.

(a) (10 pts) Represent Matching Pennies (Table 4) as an action-graph game. What is the exact number of values required to represent the utility function of each agent? In your own words, describe the difference between the normal-form and action-graph representations of Matching Pennies.

Solution. The action graph is shown below. For each agent, **2** values required to represent the utility function. Difference: In Matching Pennies, since there are only two agents and two actions for each agent, normal-form and action-graph are very similar in general. We can clearly check payoff of each agent in the normal-form representation, while we can check the dependence of each action in the action-graph representation. I think in this case(only 2 agents, 2 actions of each), normal-form representation is better.



$$\text{Player 1:} \quad u_H(\#H, \#T) = \begin{cases} 1 & \text{if } \#H=2, \#T=0 \\ -1 & \text{if } \#H=1, \#T=1 \end{cases}$$

$$u_T(\#H, \#T) = \begin{cases} 1 & \text{if } \#H=0, \#T=2 \\ -1 & \text{if } \#H=1, \#T=1 \end{cases}$$

$$\text{Player 2:} \quad u_H(\#H, \#T) = \begin{cases} -1 & \text{if } \#H=2, \#T=0 \\ 1 & \text{if } \#H=1, \#T=1 \end{cases}$$

$$u_T(\#H, \#T) = \begin{cases} -1 & \text{if } \#H=0, \#T=2 \\ 1 & \text{if } \#H=1, \#T=1 \end{cases}$$

□

(b) (10 pts) Prove directly (i.e., without appealing to the fact that congestion games always have a pure-strategy Nash equilibrium) that Matching Pennies cannot be represented as a congestion game.

Solution. In Matching Pennies, the cost of each resource is **NOT** depends on the number of other agents who select it, which contradicts with the definition of congestion game. □