COS 445 - Midterm

Due online Monday, March 10th at 11:59 pm

- All problems on this exam are *no collaboration* problems.
- You *may not* discuss any aspect of any problems with *anyone except for the course staff*.
- You *may not* consult any external resources, the Internet, etc.
- You *may* consult the course lecture notes on Ed, any of the five course readings, past Ed discussion, or any notes directly linked on the course webpage (e.g. the cheatsheet, or notes on linear programming).
- You *may* discuss the test with the course staff, but **we will only answer clarification questions and will not give any guidance or hints**. You should feel free to ask any questions and let us judge whether or not to answer, but just know that we may choose to politely decline to answer. We may choose to answer questions with a response of "I'm sorry, but I'm not comfortable answering that question," or "it is within the scope of the exam for you to answer that question yourself" (or some variant of these).
- If you choose to ask a question on Ed, *ask it privately*. We will maintain a pinned FAQ for questions that are asked multiple times (please also reference this FAQ).
- Please upload each problem as a separate file via codePost, as usual.
- You *may not* use late days on the exam. You must upload your solution by March 10th at 11:59pm. If you are working down to the wire, upload your partial progress in advance. **There is no grace period for the exam**. In case of a true emergency where you cannot upload, email smweinberg@princeton.edu and mbraverm@princeton.edu your solutions asap.
- If you miss the codePost deadline, we will not completely ignore your submission, but we will apply a **substantial** deduction before grading it. Please make sure you have something submitted by the deadline, and take into account that the server may be overloaded or sluggish near the end.
- There are *no exceptions, extensions, etc.* to the exam policy (again, in case of a *truly exceptional* circumstance, you should reach out to your residential dean and have them contact us).

Problem 1: Matching, Voting, Games, and LPs (40 points)

For each of the 4 problems below: **unless otherwise specified**[1] you do not need to show any work and can just state the answer. However, if you simply state an incorrect answer with no justification, we cannot award partial credit. You are encouraged to provide a **very brief** outline/justification in order to receive partial credit in the event of a tiny mistake. For example, we will award very significant partial credit if you clearly execute the correct outline, but make a mistake in implementation.

Part a: Stable Matchings (10 points)

Four students Alice, Bob, Claire and David are applying to summer internships at Apple, Bell Labs, Capital One and Dell (all of which need exactly one intern). Here are their preferences (listed in decreasing order).

- Alice: Capital One $\succ$ Bell Labs $\succ$ Apple $\succ$ Dell.
- Bob: Apple $\succ$ Bell Labs $\succ$ Dell$\succ$ Capital One.
- Claire: Capital One $\succ$ Dell $\succ$ Apple $\succ$ Bell Labs.
- David: Dell $\succ$ Apple $\succ$ Bell Labs $\succ$ Capital One.

and the companies preferences:

- Apple: David $\succ$ Alice $\succ$ Claire $\succ$ Bob.
- Bell Labs: Claire $\succ$ Bob $\succ$ Alice $\succ$ David.
- Capital One: David $\succ$ Claire $\succ$ Alice $\succ$ Bob.
- Dell: Bob $\succ$ David $\succ$ Claire $\succ$ Alice.

Find the stable matching that results from student-proposing deferred acceptance. A reminder of the Deferred Acceptance algorithm is the Lecture Stable Matchings I.

Part b: Voting Rules (10 points)

A town of 20 voters is holding an election between candidates Alice, Bob and Carol.

- 8 of the voters prefer Alice $\succ$ Bob $\succ$ Carol
- 7 of the voters prefer Carol $\succ$ Bob $\succ$ Alice
- 5 of the voters prefer Bob $\succ$ Carol $\succ$ Alice

State the winning candidate selected by each of the following voting rules: Borda, IRV, Plurality. A reminder of these three voting rules is in Lecture Voting Theory I.

[1] If otherwise specified, you should follow the otherwise specifications.

Part c: Game Theory (10 points)

Find a Nash equilibrium of the following game and state the expected payoff for both players. A definition of Nash equilibrium can be found in Lecture Game Theory II.

Player X, the row player, chooses between actions x_1 and x_2. Player Y, the column player, chooses between actions y_1 and y_2. The first number in each box denotes the payoff to X, and the second number is the payoff to Y. For example, if X plays action x_1 and the column player plays action y_1, then X gets payoff 3 and Y gets payoff 2.

	y_1	y_2
x_1	(3,2)	(1,8)
x_2	(0,6)	(4,4)

Part d: Linear Programming (10 points)

Write the dual of the following LP. **You do not need to solve the LP. You only need to write the dual.** A reminder of LP duality is in Lecture Linear Programming.

Maximize $6x + 2y$, such that:

- $4x + 3y \leq 10$.
- $7x + 2y \leq 6$.
- $x, y \geq 0$.

Problem 2: Build your own game (40 points)

In this problem, you are asked to design a game with several properties. You will receive partial credit for designing a game that satisfies any subset of the properties **(but you may only submit one game. I.e. you cannot design 4 games, each satisfying a single property)**. Recall also the following definitions:

Definition 1 (Weakly Dominate)**.** *A strategy a weakly dominates strategy b for player i if strategy a always yields at least as much payoff as b for player i, no matter what strategy the other players use, and also there exists a strategy profile for the other players such that strategy a yields strictly more payoff than strategy b against this strategy profile. A strategy is weakly dominant for player i if it weakly dominates all other strategies for player i.*

Definition 2 (Iterated Deletion of Dominated Strategies)**.** *Start from an arbitrary game with n players and m actions each. Let strategy a weakly dominate strategy b for player i. Then one* deletion of a dominated strategy *simply deletes the action b from player i. Now player i has only $m-1$ available actions, and all definitions of domination are updated (since player i can no longer play strategy b). Intuitively, think of this as all players reasoning that player i will never play b since it is weakly dominated).*

Iterated deletion of dominated strategies is simply repeating this operation until no weakly dominated strategies remain. If each player has only a single remaining strategy, we say that the game is solvable *by iterated deletion of dominated strategies, and we say that iterated deletion of dominated strategies predicts that each player will play their only remaining response.*

Finally, if the game is solvable by iterated deletion of dominated strategies, and the resulting strategy for each player i is s_i, we say that $(s_1, \ldots, s_n)$ results *from iterated deletion of dominated strategies.*

Specify the payoff matrix for a game with two players and two actions each such that:[2]

(i) The row player *does not* have a **weakly dominant** pure strategy. *(10 points)*

(ii) The column player has a **weakly dominant** pure strategy. Furthermore, after deleting the column player's **weakly dominated** pure strategy, the row player now has a **weakly dominant** pure strategy. (In other words, the game is solvable by iterated deletion of weakly dominated strategies). *(10 points)*

(iii) The game has *exactly two* pure Nash equilibria. *(10 points)*

(iv) The pure Nash equilibrium resulting from iterated deletion of dominated strategies is *strictly worse* for both players than the other pure Nash equilibrium (which does not result from iterated deletion of dominated strategies). That is, each player receives strictly less payoff in the pure Nash equilibrium resulting from iterated deletion of dominated strategies than in the other pure Nash equilibrium.*(10 points)*

For each property, provide a brief justification of why your game satisfies it.

Note: It is OK if your game has a mixed Nash equilibrium that is not pure, and also OK if your game does not have a mixed Nash equilibrium that is not pure (and also OK if your game has any number of mixed Nash equilibria). You do not have to reason about mixed Nash equilibria at all in this problem for full credit.

[2] You do not need to provide any flavor text to explain the game. You just need to design the payoff matrix.

Problem 3: Positive Responsiveness and Strategyproofness (50 points)

Note: You may NOT cite any results from the course materials for this problem. Your proof for this problem must be entirely self-contained (that is, you must prove any claim that you make, and may not cite the fact that it is stated/proved in a lecture/textbook). A "self-contained proof" is one that you came up with on your own, or at minimum one that you could reproduce on your own without any sources.[3]

Recall the following definitions:

Definition 3. *A voting rule F is strategyproof if no voter can ever be strictly happier by lying. Formally, F is strategyproof if for all voters i, all true preferences $\succ_i$, and all possible lies $\succ_i'$, and all possible votes of other voters $\vec{\succ}_{-i}$, $F(\succ_i; \vec{\succ}_{-i}) \succeq_i F(\succ_i'; \vec{\succ}_{-i})$.*[4]

Definition 4 (Unanimous)**.** *A voting rule F is unanimous if for all candidates a, whenever all voters select a as their favorite candidate, F outputs a. Formally, F is unanimous if whenever there exists a candidate a such that $a \succ_i b$ for all i and all $b \neq a$, then $F(\succ_1, \ldots, \succ_n) = a$.*

Here is a new definition:

Definition 5. *A candidate b is* negatively positioned *for votes $\succ_1, \ldots, \succ_n$ if there exists another candidate a such that $a \succ_i b$ for all i.*

A voting rule F is positively responsive *if for all $\succ_1, \ldots, \succ_n$, $F(\succ_1, \ldots, \succ_n)$ is* not *negatively positioned for $\succ_1, \ldots, \succ_n$ (that is, F never outputs a negatively positioned candidate).*

Prove that every strategyproof, unanimous voting rule is positively responsive.

If you fully solve the stated problem, you will get full credit. Below are some related lemmas to the problem that you might prove for partial credit (depending on how you approach the problem, you may or may not find these lemmas to be useful towards your proof).

Note: Neither of the two lemmas below are "hints" – they are just intended as concrete suggestions for partial credit in case you're stuck, and as inspiration to get your gears going. You may find that your ultimate solution makes use of neither part.

Clarifying Grading: For this problem, there is one quality of progress mark, and one quality of presentation mark. The parts below are suggestions for concrete things you could try to prove in order to improve your quality of progress mark. Inside the parentheses is the highest mark you could earn for fully solving that optional part and nothing else.[5]

Optional Part a: Participation (Significant Concrete Partial Progress)

Let F be a strategyproof voting rule. Prove that F is *Participatory*, defined below:

[3] Therefore, you may not bypass the prohibition on citing course materials by including a re-written proof of a result from the notes.

[4] Here, we have used the notation $x \succeq_i y$ to denote "either $x \succ_i y$ or $x = y$." This is the same as saying $x \succeq_i y$ whenever $y \not\succ_i x$.

[5] There is not a precise formula for combining multiple "some concrete partial progress"es – these are all just suggestions for things to try if you're stuck.

Definition 6. *A voting rule F is **Participatory** if for all preference lists $\succ_1, \ldots, \succ_n$, $\succ'_1, \ldots, \succ'_n$ such that (1) $F(\succ_1, \ldots, \succ_n) = a$, and (2) for all voters i and all candidates $b \neq a$, $a \succ_i b \Rightarrow a \succ'_i b$ (for all voters i and candidates b, if they preferred a to b under $\succ_i$, they still prefer a to b under $\succ'_i$), then (3) $F(\succ'_1, \ldots, \succ'_n) = a$ as well.*

Intuitively, a voting rule satisfies Participation if a voter cannot harm themselves by "participating" (in a manner that is consistent with their true preferences towards the winning candidate).

Optional Part b: Thank You, Lecture (Some Concrete Partial Progress)

Prove that every strategyproof, unanimous voting rule is positively responsive, *but you may cite without proof any claim in Lecture 5 or 6* (you must still cite it exactly as stated). To clarify, if you solve Problem 3 *while citing a claim in Lecture 5 or 6 without proof*, you could earn up to "Some Concrete Partial Progress" on the generic rubric for doing so.[6]

[6]For full credit, you must solve Problem 3 without citing any results from Lecture 5 or 6 or the course material, as stated in the problem description.

Problem 4: Optimized Matchings (70 points)

For this problem, there are n students and n universities, each with 1 slot, and $n > 25$.

Definition 7 (k-demand)**.** *A matching is k-demand if every student is matched to one of their top k choices, and every university is also matched to one of their top k choices.*

We say that a matching is k-demand for students if every student is matched to one of their top k choices, and k-demand for universities if every university is matched to one of their top k choices (and therefore a matching is k-demand if it is both k-demand for students and k-demand for universities).

Given as input preferences $\succ_1^s, \ldots, \succ_n^s$ of n students over the n universities, and preferences $\succ_1^u, \ldots, \succ_n^u$ of n universities over the n students, design an algorithm that outputs a stable matching that is 25-demand, whenever one exists. Prove that your algorithm is correct. For full credit, your algorithm must run in time polynomial in n. For a proof to be 'complete', it must show that the algorithm is correct, but does not need to analyze its runtime.[7]

If you fully solve the stated problem, you will get full credit. Below are some related lemmas to the problem that you might prove for partial credit (depending on how you approach the problem, you may or may not find these lemmas to be useful towards a complete proof).

Note 1: Depending on how you approach the problem, you may or may not find it helpful to cite concrete lemmas from Lecture 1 or Lecture 2. If you choose to do so, you should make absolutely sure that you are applying the lemma *exactly* as stated and proved in the lecture.

Note 2: None of the three lemmas below are "hints" – they are just intended as concrete suggestions for partial credit in case you're stuck, and as inspiration to get your gears going. You may find that your ultimate solution makes use of none of these parts.

Clarifying Grading: For this problem, there is one quality of progress mark, and one quality of presentation mark. The parts below are suggestions for concrete things you could try to prove in order to improve your quality of progress mark. Inside the parentheses is the highest mark you could earn for fully solving that optional part and nothing else.[8]

Optional Part a (Some Concrete Partial Progress)

Consider running Student-Proposing Deferred Acceptance on input $\succ_1^s, \ldots, \succ_n^s, \succ_1^u, \ldots, \succ_n^u$, and let M^s denote the output. Similarly, consider running University-Proposing Deferred Acceptance on input $\succ_1^s, \ldots, \succ_n^s, \succ_1^u, \ldots, \succ_n^u$, and let M^u denote the output. Prove that if M^s is not 25-demand for students, then no stable matching is 25-demand for students. Similarly, prove that if M^u is not 25-demand for universities, then no stable matching is 25-demand for universities.

[7] Most reasonable algorithms, including ones we've seen in class, run in polynomial time. The purpose of this requirement is to rule out 'trivial' brute-force solutions, such as one iterating over all $n!$ matchings and checking whether each one satisfies the condition.

[8] There is not a precise formula for combining multiple "some concrete partial progress"es – these are all just suggestions for things to try if you're stuck.

Optional Part b (Significant Concrete Partial Progress)

Definition 8. *For a given input $\succ_1^s, \ldots, \succ_n^s$, $\succ_1^u, \ldots, \succ_n^u$, consider adding one additional student and university in the following manner, which we'll refer to as an (i,k)-rural addition.*

- *The "rural" student $n+1$ has $(\succ_{n+1}^s)'$ prefer university i first, followed by university $n+1$, followed by the remaining universities (in arbitrary order).*
- *The "rural" university $n+1$ has $(\succ_{n+1}^u)'$ prefer student i first, followed by student $n+1$, followed by the remaining students (in arbitrary order).*
- *Student i updates its preferences by inserting university $n+1$ into its $(k+1)^{st}$ favorite slot, and shifting all universities previously ranked $k+1$ or below down by one slot. Refer to this as $(\succ_i^s)'$.*
- *University i updates its preferences by inserting student $n+1$ into its $(k+1)^{st}$ favorite slot, and shifting all students previously ranked $k+1$ or below down by one slot. Refer to this as $(\succ_i^u)'$.*
- *All other students j update their preferences by adding university $n+1$ as its least-favorite university. Refer to this as $(\succ_j^s)'$.*
- *All other universities j update their preferences by adding university $n+1$ as its least-favorite student. Refer to this as $(\succ_j^u)'$.*

We refer to $(\succ_1^s)', \ldots, (\succ_{n+1}^s)', (\succ_1^u)', \ldots, (\succ_{n+1}^u)'$ as the (i,k)-rural addition of $\succ_1^s, \ldots, \succ_n^s$, $\succ_1^u, \ldots, \succ_n^u$.

For a given matching M on n students to n universities, let M' modify M by simply adding the match between student $n+1$ and university $n+1$. Prove that if M is stable for $\succ_1^s, \ldots, \succ_n^s$, $\succ_1^u, \ldots, \succ_n^u$ *and student i gets one of their top k choices in M and university i gets one of their top k choices in M*, then M' is stable for the (i,k)-rural addition of $\succ_1^s, \ldots, \succ_n^s$, $\succ_1^u, \ldots, \succ_n^u$.

Optional Part c (Some Concrete Partial Progress)

Design an algorithm that outputs a stable matching that is 1-demand, whenever one exists. Prove that your algorithm is correct.