COS 445 - PSet 1

Due online Monday, February 7th at 11:59 pm

Instructions:

- Some problems will be marked as no collaboration problems. This is to make sure you have experience solving a problem start-to-finish by yourself in preparation for the midterms/final. You cannot collaborate with other students or the Internet for these problems (you may still use the referenced sources and lecture notes). You may ask the course staff clarifying questions, but we will generally not give hints.
- Submit your solution to each problem as a **separate PDF** to codePost. Please make sure you're uploading the correct PDFs![1] If you collaborated with other students, or consulted an outside resource, submit a (very brief) collaboration statement as well. Please anonymize your submission, although there are no repercussions if you forget.
- The cheatsheet gives problem solving tips, and tips for a "good proof" or "partial progress": `http://www.cs.princeton.edu/~smattw/Teaching/cheatsheet445.pdf`.
- Please reference the course collaboration policy here: `http://www.cs.princeton.edu/~smattw/Teaching/infosheet445sp22.pdf`.

[1] We will assign a minor deduction if we need to maneuver around the wrong PDFs. Please also note that depending on if/how you use Overleaf, you may need to recompile your solutions in between downloads to get the right files.

Problem 1: Unique Stable Matchings (20 points, no collaboration)

Let there be n students and n universities, each with capacity one. Recall that a list of $2n$ preferences, $\vec{\succ} = \langle \succ_1, \ldots, \succ_{2n} \rangle$ defines a stable matching instance. Consider an instance $\vec{\succ}$ where every student has exactly the same (strict) preference ordering over universities. That is, for all students i, j, $\succ_i = \succ_j$, but you may not make any assumptions on the university preferences.

Prove that, in any instance $\vec{\succ}$ where every student has the same strict preferences, that there is a unique stable matching for $\vec{\succ}$. That is, there is a single matching M that is stable for $\vec{\succ}$, and any other matching $M' \neq M$ is unstable for $\vec{\succ}$.

Problem 2: Other matching algorithms (40 points)

In this problem, we'll consider the behavior of algorithms other than deferred acceptance for the standard two-sided stable matching problem. For each of the following two algorithms, and each of the following four statements, prove the statement or find (and analyze) a counterexample. You should assume that there are n students and n universities, each university has one slot (and each student wants one university), and all preferences are strict.

i. The algorithm always outputs a Pareto-optimal matching.

ii. The algorithm always outputs a stable matching.

iii. The algorithm is incentive-compatible for each student.

iv. The algorithm is incentive-compatible for each university.

Part a: Serial dictatorship (20 points)

For better or worse, the following algorithm treats the stable matching problem like a housing lottery:

1. Initialize a temporary matching $M := \emptyset$.
2. Pick the lexicographically next student s who is unmatched in M.
3. Match s to her favorite university that isn't matched in M.
4. Repeat from step 2 until all students are matched.

Part b: Weighted matching (20 points)

Noticing the superficial similarities between the stable matching problem and bipartite matching, we might be tempted to turn the former into an instance of the latter. We might end up with an algorithm like this:

1. Define $R_s(u)$, the rank of u for s, to be the number of universities that s prefers to u. Similarly, define $R_u(s)$, the rank of s for u, to be the number of students that u prefers to s.
2. Define the weight w_{su} of the edge (s, u) to be $R_s(u) + R_u(s)$.
3. Output a minimum-weight perfect matching in the complete bipartite graph with edge weights $\vec{w}$.[2]

[2]How to break ties is entirely up to you: If you choose to write a proof, you may break ties however you like. If you choose to write a counterexample, you may also break ties however you like.

Problem 3: The stability of greed (40 points)

Suppose there are n students (who each want one university), and n universities with one slot each. Now, let A be an $n \times n$ matrix with positive real entries ($A_{ij} \in \mathbb{R}_+$ for all i, j), that additionally satisfies the following conditions:

- For all rows i, the n entries of A in row i are all distinct ($A_{ij} \neq A_{ik}$ for all i, and $j \neq k$).
- For all columns j, the n entries of A in column j are all distinct ($A_{ij} \neq A_{kj}$ for all j, and $i \neq k$).

We now define a stable matching instance so that the students' and universities' preferences are formed based on A.

- For university i, let i_k denote the column of the k^{th} largest entry of row i. Then university i prefers students in the following order: $i_1 \succ i_2 \succ \ldots \succ i_n$.
- For student j, let j_k denote the row of the k^{th} largest entry of column j. Then student j prefers universities in the following order : $j_1 \succ j_2 \ldots \succ j_n$.

So for example, if $A = \begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$, then university 1 prefers student 1 to 2, and university 2 prefers student 2 to 1. Similarly, student 1 prefers university 1 to 2 and student 2 prefers university 2 to 1.

Part a: Greedy works! (15 points)

Consider the following greedy algorithm, henceforth referred to as GREEDY:

1. Let $a_{i,j}$ be the largest entry in the matrix A (in case of a tie, choose the entry with the smaller i).
2. Match university i with student j.
3. Update A by setting $a_{i,k} := -1$ for all k, and $a_{k,j} := -1$ for all k (i.e. remove row i and column j from future consideration because all "untouched" entries are positive).
4. Repeat until all entries in A are -1.

Show that the matching output by GREEDY is stable. You do not need to prove that GREEDY outputs a matching, that it terminates, or analyze its runtime.

Part b: Greedy is all that works! (15 points)

Show that the matching output by GREEDY is the unique stable matching. In other words, prove that all other matchings are not stable.

Part c: But it's not perfect (10 points)

Provide an example of an A such that the matching returned by GREEDY is not the maximum weight matching of the underlying bipartite graph. More specifically, if M denotes a matching from $[n]$ to $[n]$, define the weight of M with respect to A as $\sum_i A_{i,M(i)}$. Construct an A such that there exists a matching M whose weight exceeds the weight of the matching returned by GREEDY.

Extra Credit: Almost Unique Stable Matchings

Recall that extra credit is not directly added to your PSet scores, but will contribute to your participation. Some extra credits are **quite** challenging and will contribute significantly.

Consider an instance with n students and n universities where student preferences are uniformly random, and university preferences are arbitrary. However, instead of a full preference ordering over all n universities, each student truncates their preferences at the top $c = O(1)$ universities (that is, they prefer to be unmatched rather than partner with a school outside their top c).

Say that a university is uniquely stable for this instance if they have the same partner in all stable matchings (where "unmatched" counts as a partner). Prove that the expected number of uniquely stable universities is $n - o(n)$.[3]

This is a long problem, and the following hints break down the key steps. If you can clearly state and prove concrete steps (e.g. clearly state claims suggested by some of the hints, and prove them), you will get partial extra credit.

Hint 1: You may want to prove the following fact first. Let M, M' be any two stable matchings. Then every student who is matched in M is also matched in M' (and vice versa). Every university that is matched in M is also matched in M' (and vice versa).

Hint 2: You may also want to prove the following: Let M be output by student-proposing deferred acceptance (i.e. each student stops applying if they are rejected by all of their top c schools), and let $M(u) = s$. Now consider modifying u's preferences by "blacklisting" s and all s' that u likes less than s. That is, u declares that they would rather be unmatched that matched to s or anyone below s. Then any matching M' where $M'(u) = s' \neq M(u)$ is stable for the new preferences if and only if it is stable for the original preferences.

Hint 3: You may next want to prove the following fact using Hints 1 and 2. Let M be output by student-proposing deferred acceptance (where each student only proposes to a university to which they apply), and let $M(u) = s$. Now consider modifying u's preferences by "blacklisting" s and all s' that u likes less than s. That is, u declares that they would rather be unmatched that matched to s or anyone below s. Let M' denote the matching output by student-proposing deferred acceptance with this modified preference (and all others the same). If u is unmatched, then u is matched to s in every stable matching.

Hint 4: To start wrapping up, you may want to use the fact that Student-Proposing Deferred Acceptance outputs the same matching, independent of the order in which students propose (this is a corollary of a theorem from lecture). In particular, you may want to choose the order in which students propose to make use of the earlier hints.

Hint 5: Finally, you may use the following fact without proof:[4] imagine throwing k balls into n bins uniformly at random without replacement, and then repeating this procedure n times independently (so we pick n uniformly random lists of k distinct bins). Then with probability $1 - e^{-\Omega(n)}$, at least $n \cdot e^{-k}/4$ bins are empty.

[3] Observe that this means it barely matters which side proposes in this model because almost everyone has the same partner regardless.

[4] This fact is oddly stated, because it is tailored to this problem to remove the need for calculations.