Announcements

- We’ll be going over midterm solutions this weekend at the normal times and locations.
- Remember to complete the weekly quiz by Sunday 9pm.

Definitions

- A matching in $G$ is a subset of $G$’s edges which share no vertices.
  A maximal matching is one which isn’t a subset of any other matching.
  A maximum matching is a matching which is at least as large as any possible matching.
  A perfect matching is a matching such that every vertex is contained in one of its edges.

- An alternating path (with respect to some matching $M$) is one which alternates between edges in $M$ and edges not in $M$.
  An augmenting path is an alternating path that begins and ends with vertices unmatched in $M$.

- An unstable pair is a pair who prefer each other to their assigned partners.
  A stable matching is a perfect matching (includes all vertices) which contains no unstable pairs.

- Gale Shapley algorithm on sets $A$ (men) and $B$ (women), as follows...
  While there is a man $m \in A$ who is unmatched:
  
  (a) Let $w \in B$ be the highest ranked woman in $m$’s list whom he hasn’t proposed to yet.
  
  (b) If $w$ is unmatched or prefers $m$ to her current match, match $w$ and $m$.

Corridor’s Theorem

(a) Recall and prove Hall’s Theorem, restated below:

Let $G = (X, Y, E)$ be a bipartite graph, and for any $S \subseteq X$ let $N(S) = \{y \in Y \mid \exists x \in S \text{ s.t. } \{x, y\} \in E\}$, i.e. the neighbors of $S$. Then $G$ has a matching covering all of $X$ iff for every $S \subseteq X$, $|S| \leq |N(S)|$.

See Chapter 8 of the notes. Alternatively, one can also prove this via induction, casing on whether there exists a non-empty, proper subset $S$ of $X$ satisfying $|N(S)| = S$.

A Misogynist Algorithm

(a) Prove that the Gale-Shapley algorithm always matches every guy with his best valid partner. That is, show that every guy prefers the girl he is paired with by the Gale-Shapley algorithm at least as much as any girl he is paired with in any other stable matching.
(b) Prove that the Gale-Shapley algorithm always matches every girl with her worst valid partner. That is, show that in any other stable matching, each girl is paired with a guy she likes at least as much as the one she is paired with by Gale Shapley.

(Extra) Soulmates

Call a man $m$ and a woman $w$ “soulmates” if they are paired with each other in every stable matching.

(a) Describe a polynomial-time algorithm that, given a man $m$ and woman $w$, determines whether they are soulmates.

Algorithm Description:
Let $\text{Gale-Shapley}(A, B)$ be a function to simulate the Gale Shapley algorithm to match elements of $A$ with elements in $B$ so that the resulting matching is optimal to the elements of $A$.
We first run Gale-Shapley in a male-optimal (female-pessimal) way, i.e. match men ($M$) to women ($W$).
Next, we swap the role of men and women and run Gale-Shapley in a female-optimal (male-pessimal) way.
If $m$ and $w$ were paired in both cases, we say that they’re soulmates, otherwise we say that they aren’t.

Correctness:
Lemma 1: (From Notes) If $a \in A$ and $b \in B$ are paired by running $\text{Gale-Shapley}(A, B)$, then $b$ is $a$’s best valid partner and $a$ is $b$’s worst valid partner.
If $m$ and $w$ are not paired in either of these stable matchings, $m$ and $w$ aren’t soulmates because there exists a stable matching where they aren’t paired.
If $m$ and $w$ are paired in both of these stable matchings, we know that $w$ is $m$’s best valid partner and also $m$’s worst valid partner (from Lemma 1). Since $w$ is $m$’s best and worst valid partner, she must be his only valid partner, so $m$ and $w$ must be paired together in every stable matching.

Runtime analysis:
Since we make two calls to Gale-Shapley, the runtime of our algorithm will be twice the runtime of Gale-Shapley (plus a constant for checking if $m$ and $w$ are paired). Since Gale-Shapley is poly-time, our algorithm is poly-time as well.

(b) Give a polynomial time algorithm to determine if an instance of the stable matching problem has a unique stable matching.
Algorithm Description:
Let $\text{Gale-Shapley}(A, B)$ be a function to simulate the Gale Shapley algorithm to match elements of $A$ with elements in $B$ so that the resulting matching is optimal to the elements of $A$.
We first run Gale-Shapley in a male-optimal (female-pessimal) way, i.e. match men ($M$) to women ($W$).
Next, we swap the role of men and women and run Gale-Shapley in a female-optimal (male-pessimal) way.
For a matching to be stable, both matchings must be exactly the same, i.e. for each matched pair, $(m,w)$, $m$ must be both $w$’s best valid partner as well as worst valid partner, and vice versa.

Correctness:
By the previous part, if some pair $m$ and $w$ are matched in the pairing output by male-optimal (female-pessimal) Gale-Shapley, and also matched in the pairing output by female-optimal (male-pessimal) Gale-Shapley, $m$ and $w$ must be soulmates.
If every couple in the matching output by male-optimal Gale-Shapley is also a couple in the matching output by female-optimal Gale-Shapley, every couple in the matching output by Gale-Shapley are soulmates. This implies every one of these couples must be together in every stable matching. Thus, the only stable matching is the one where all these couples are together.

Runtime Analysis:
As before, our runtime is just twice the runtime of Gale-Shapley, which is polynomial. Checking if the two matchings are the same can be done in linear time. Thus, the total runtime is polynomial.

(Extra) Counting in a Couple Ways

(a) Find, with proof, the maximum possible number of perfect matchings in a graph on $n$ vertices.

Any matching in a graph on $n$ vertices will also be a matching in the complete graph on $n$ vertices, so it suffices to find the number of perfect matchings in $K_n$. If $n$ is odd, there are no perfect matchings, since any way of pairing the vertices must leave some vertex unpaired.
If $n$ is even, there are $n - 1$ ways to choose a match for the first vertex, then $n - 3$ ways to choose a partner for the next unmatched vertex, then $n - 5$ ways to choose a partner for the next unmatched vertex after that, and so on, each yielding different matchings. Thus, the total number of matchings is $\prod_{i=1}^{n/2} (2i - 1) = \frac{n!}{2^{n/2}(n/2)!}$.

(b) Find, with proof, the maximum possible number of perfect matchings in a bipartite graph on $n$ vertices.

Again, a graph on $n$ vertices can only have a perfect matching if $n$ is even. Furthermore, a bipartite graph can only have a perfect matching if the two partitions have the same size, so in this case, the partitions must be of size $n/2$. Once we fix the partitions, any perfect matching on a graph with these partitions will be a perfect matching in the complete graph with these partitions, so it suffices to count the number of perfect matchings in $K_{n/2, n/2}$.
This time there’s $\frac{n}{2}$ choices for the match of the first vertex, then $\frac{n}{2} - 1$ choices for the match of the second vertex, and so on, since each match we make decreases the number of vertices on each side by 1. Thus, the total number of perfect matchings is $\frac{n!}{2^{n/2} (n/2)!}$.

(c) Find a way to construct an instance of the stable marriage problem with $n$ men and $n$ women which has at least $n$ stable matchings (tight bounds on the number of stable matchings for $n$ pairs of men and women are not known).
Label the men $m_0$ through $m_{n-1}$ and the women $w_0$ through $w_{n-1}$. Let the preference list of $m_i$ be $(w_i, w_{i+1}, \ldots, w_{n-1}, w_0, w_1, \ldots, w_{i-1})$, and the preference list of $w_i$ be $(m_{i+1}, m_{i+2}, \ldots, m_{n-1}, m_0, m_1, \ldots, m_i)$. Note that, for any $i$ and $j$, $m_i$ has $w_j$ at position $j - i \mod n$ on his preference list, and $w_j$ has $m_i$ at position $i - j - 1 \mod n$ on her preference list. Thus, each woman occurs at a different place on each man’s preference list (and vice-versa), so each man has a different $k$th favorite woman. We claim that for any $0 \leq k < n$, pairing each man with his $k$th favorite woman gives a stable matching. To see this, note that $m_i$’s $k$th favorite woman is $w_{i+k}$, and the only women that $m_i$ prefer’s to her are $w_i$ through $w_{i+k-1}$. These women, however, rank $m_i$ in positions from $n - 1$ to $n - k$ on their preference lists (since $i - (i + x) - 1 = -x - 1 = n - x - 1$), while they rank their current partners in position $n - k - 1$ on their preference lists (because $(j - k) - j - 1 = -k - 1 = n - k - 1$), and since $0 \leq k \leq n - 1$, this is a higher position on the list, so all these women prefer their current partners to $m_i$, so there are no unstable pairs involving $m_i$ for any $i$, so the matching is stable. Thus, we have found a set of preference lists with at least $n$ distinct stable matchings.

**Bonus** Pharaoh’s Theorem

(a) Given $G = (V, E)$ and $S \subseteq V$, let $G \setminus S$ be the graph formed by removing the vertices in $S$ and all the edges incident to them. Also, define $o(G)$ to be the number of connected components in $G$ of odd size. Then prove the following theorem:

Let $G = (V, E)$ be any graph. Then $G$ has a perfect matching iff for every $S \subseteq V$, $o(G \setminus S) \leq |S|$. 
Be warned that this is a fairly involved proof.

\(\Rightarrow\) Fix a perfect matching \(M\) of \(G\), and let \(S \subseteq V\) be arbitrary. Note that for every odd component \(C\) of \(G \setminus S\), there must exist an edge in \(M\) connecting a vertex in \(C\) with a vertex in \(S\); this is because every edge incident to \(C\) either connects two vertices in \(C\) or one in \(C\) and one in \(S\), and since \(C\) is of odd size there must be a vertex in \(C\) whose matched vertex is not in \(C\). Hence, every odd component corresponds to at least one vertex in \(S\). Further, since \(M\) is a matching, no vertex in \(S\) can correspond to more than one odd component in \(G \setminus S\), and so \(o(G \setminus S) \leq |S|\), as desired.

\(\Leftarrow\) We’ll prove the contrapositive, so suppose that \(G\) has no perfect matching. Without loss of generality, suppose also that \(G\) is edge-maximal with respect to not having a perfect matching, i.e. that \(G + e\) contains a perfect matching for any edge \(e\); we may make this assumption because for any \(S \subseteq V\), every odd component of \((G + e) \setminus S\) is the union of components of \(G \setminus S\), at least one of which must be odd. Also, note that if \(G\) is of odd order then taking \(S = \emptyset\) completes the proof, so we may assume that \(G\) is of even order.

Observe that if there does exist a set \(S\) such that \(o(G \setminus S) > |S|\), then by edge-maximality every component of \(G \setminus S\) is complete and every vertex in \(S\) is adjacent to every vertex in \(G \setminus S\); the addition of these edges does not change \(o(G \setminus S)\), and by the forwards direction this means there is no perfect matching. But if there is such a set \(S\) exists and \(G\) has even order, then it is easy to recover a perfect matching by matching each vertex in \(S\) to a vertex in \(G \setminus S\), and then matching remaining pairs within each component arbitrarily. So it suffices to show that there exists a set \(S\) satisfying the above condition.

Towards this, set \(S\) to be the vertices in \(G\) adjacent to every other vertex. If \(S\) satisfies the given condition, then we’re done; otherwise, there exists a component of \(G \setminus S\) containing two vertices \(a \sim a'\) (by our construction of \(S\)). Since they’re in the same component, there exists a path from \(a\) to \(a'\) entirely within the component; let \(b\) and \(c\) be the next vertices on an arbitrary such path of minimal length. Then \(ab\) and \(bc\) are both edges in \(G\), but \(ac\) is not. Furthermore, since \(b \notin S\), there exists a vertex \(d\) in \(G \setminus S\) such that \(b \sim d\). By edge-maximality, observe that \(G + ac\) has a perfect matching \(M_1\) and \(G + bd\) has a perfect matching \(B_2\). Using these two matchings, we’ll now construct a perfect matching in \(G\) to show that this case cannot happen.

\[
\begin{array}{c}
\text{\includegraphics[width=0.3\textwidth]{diagram.png}}
\end{array}
\]

Let \(P\) be a maximal path in \(G\) starting at \(d\) starting with an edge from \(M_1\) and alternating edges between \(M_1\) and \(M_2\). If the last edge of \(P\) is an edge from \(M_1\), then observe that that the last vertex must be \(b\). At any vertex that’s not \(b\) or \(d\), we could take an outgoing edge from \(M_2\) to continue the path, and we cannot end at \(d\). In this case, let \(C\) be the cycle containing \(P\) and the edge \(bd\). On the other hand, if the last edge of \(P\) is an edge from \(M_2\), then analogously we must have ended at either \(a\) or \(c\) (without loss of generality, \(a\)). Then let \(C\) be the cycle containing \(P\) and the edges \(ab\) and \(bd\). Either way, \(C\) is a cycle of even length where every other edge is in \(M_2\), and \(bd\) is the only edge in \(C\) that is not in \(G\). Taking the other half of the edges in \(G\) combined with the rest of \(M_2\) yields a perfect matching in \(G\), as desired.