Stable Matchings

In real-world settings that involve finding matchings, the problem often goes beyond simply maximizing the cardinality of the matching. Actors on both sides may have preferences for regarding who they wish to be matched with. Suppose that two vertices a and b have an edge but are not matched by a matching M (they may both be matched to other vertices and/or be unmatched). If both a and b prefer to be matched to each other over their current situation, the matching M is unstable. The famous theorem below, by Gale and Shapley, shows that with any set of preferences, there always exists a stable matching. In 2012, Lloyd Shapley and Alvin Roth received a Nobel prize for their work in matching theory (David Gale had passed away in 2008).
We call a family of linear orderings $(\leq_v)_{vinV}$ a set of *preferences*. We say that a matching M is *stable* if for every edge $e \in E \setminus M$, there is an edge $f \in M$ such that e and f share a vertex v and $e \leq_v f$.

Theorem 1. *For every set of preferences, a bipartite graph G has a stable matching.* 2.1.4 p.40

Exercise 1. *We run Gale-Shapley on a set of preference rankings by n doctors and n hospitals. The number of iterations before the algorithm converges depends on the preference rankings. What is the least number of iterations the algorithm can go through to find a stable matching? What is the most number of iterations?* 3 points

Exercise 2. *Suppose that we are given a set of preference rankings by n doctors and n hospitals. However, the preference rankings for each doctor rank only a subset of the available hospitals, the others are vetoed (the doctor would never work there). If we consider non-veto edges, then Hall's Condition is satisfied (so there exists an assignment of all doctors to hospitals they have not vetoed). Does Gale-Shapley work with the restriction that doctors may refuse to apply to some hospitals? Is it possible that there is no stable perfect matching (in which every doctor is matched to a hospital) in such a case?* 2 points

We discuss one more nice application of Hall's Theorem (which will come back again when we discuss cuts and flows). A k-factor in a graph G is a subgraph $H \subseteq G$ such that $V(H) = V(G)$ (we call this a *spanning* subgraph) and H is k-regular. A perfect matching, one that covers all the vertices of a graph G, there gives us a 1-factor. Note that they are not technically the same, as a matching is a set of edges, while a 1-factor is a subgraph.

Theorem 2. *Every regular graph of positive even degree has a 2-factor.* 2.1.5 p.41

Tutte's Theorem in Section 2.2 gives us a generalization of Hall's Theorem to perfect matchings in general, not necessarily bipartite graphs. This proof is a bit bigger, and we may revisit it later in this course. To gain some intuition for Tutte's Theorem, try to prove the following result for perfect matchings in trees. (Of course, trees are bipartite, but do not use Hall's Theorem.)

Exercise 3. *Show that a tree T has a perfect matching if and only if T is such that for every $v \in V(T)$, the graph $T - v$ has exactly one component with an odd number of vertices.* 2 points

k-connectedness

We say that a graph G is k-connected (or, k-vertex-connected) if $G \sim K_n$ or if G is connected and remains connected if the deletion of any set of at most $k-1$ vertices leaves G connected.

In other words, disconnecting G requires the removal of at least k vertices. Note that this generalizes the definition of connectedness that you are already familiar with, which is really 1-vertex-connectedness. We will start with investigating the structure of 2-connected graphs. Let H be a subgraph of a graph G. We call a path P in G an *H-path* if the endpoints of P lie in H, but all other vertices of P are outside of H. We will also call this an *ear* on H. Now, we let an *ear decomposition* be a series of subgraphs of G: $C = H_0, H_1, \ldots, H_k = G$, such that C is a cycle in G, and each H_i is obtained from H_{i-1} by attaching an ear.

Proposition 3. *Every 2-connected graph has an ear decomposition.* 3.1.1 p.59

We can find a similar result for edge-connectedness. We say that a graph G is k-edge-connected if G is connected and remains connected if the deletion of any set of at most $k-1$ edges leaves G connected.
We call a *closed ear* on H a cycle that has exactly one vertex in common with H.

Proposition 4. *Every 2-edge-connected graph has an ear decomposition that uses open and/or closed ears.*

Exercise 4. *Prove Proposition 4.* 2 points

We start a few more exercises to familiarize ourselves with vertex- and edge-connectivity. We say that G has vertex-connectivity $\kappa(G)$ if G is κ-connected but not $\kappa + 1$-connected. Similarly, we label the edge-connectivity as $\lambda(G)$.

Exercise 5. *Show that $\kappa(G) \leq \lambda(G)$ for every graph G.* 3 points

Exercise 6. *Show that $\lambda(G) \leq \delta(G)$ for every graph G.* 1 points

Exercise 7. *Suppose that G has all vertices of even degree. What do we know about the parity of $\kappa(G)$? What do we know about the parity of $\lambda(G)$?* 2 points