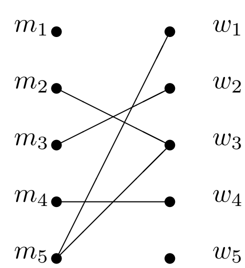
Introduction to Graph Theory

Motivation and basic definitions: In class we saw an example of computing probabilities, and we saw that a lot of information can go into these computations. So we'd like a good way to store and organize and manipulate this information. It turns out that graph theory is a great tool.

We'll need some basic definitions. A **graph** is a list of **vertices** $\{v_1, \ldots, v_n\}$ and **edges**. Each edge joins a pair of distinct vertices (so we don't consider edges that join a vertex to itself), and we write the edge joining v_i to v_j as $v_i - v_j$. Notice that we don't put a direction to this edge, so that in graph theory $v_i - v_j = v_j - v_i$. If there is an edge $v_i - v_j$ we say that the vertices v_i and v_j are **adjacent**, and we say the edge $v_i - v_j$ is **incident** to both vertices v_i and v_j. The **degree** of a vertex is the number of edges incident to it.

Sometimes you want to have a direction to the edges, that is you want to distinguish between the edge $v_1 - v_2$ and the edge $v_2 - v_1$. In this case we write edges with an arrow to indicate direction, such as $v_1 \to v_2$, and call the graph a directed graph.

Let's look at an example of how graph theory is useful. In 1994, the University of Chicago published a study called *The Social Organization of Sexuality*, in which they claimed that on average men have 74% more opposite-sex partners than women. Does this make sense? In other words, can this statistic hold over an entire population. We can figure this out by writing down a graph G. We write down a vertex for each person, and it will be convenient to split up the vertices as the collection M of male vertices and the collection W of female vertices. So the entire vertex set of G is $V = \{M, W\}$. If $m \in M$ represents a man and $w \in W$ represents a women, we draw an edge $w - m$ between them if they've been partners; these are the only edges in our graph. Part of our graph might look like this:



Now, each man m has $\deg(m)$ partners of the opposite sex, while each woman w has $\deg(w)$ partners of the opposite. This means we can compute the average number of partners of the opposite sex, for both men and women as

$$\text{avg}(M) = \frac{\Sigma_{m \in M} \deg(m)}{\#M}, \qquad \text{avg}(W) = \frac{\Sigma_{w \in W} \deg(w)}{\#W}.$$

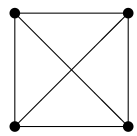
Now, if we take the ratio of these averages, we get

$$\frac{\text{avg}(M)}{\text{avg}(W)} = \frac{(\#W)\Sigma_{m \in M} \deg(m)}{(\#M)\Sigma_{w \in W} \deg(w)} = \frac{\#W}{\#M}.$$

Here we have used the fact that $\Sigma_{m \in M} \deg(m) = \Sigma_{w \in W} \deg(w)$, which holds because each edge runs from W to M, so each of these sums is just the total number of edges in our graph. We see that the University of Chicago's statistics can't hold over an entire population, unless there are many more women than men. This means the statistics say a lot more about people filling out studies than it does about sexual practices. (Namely, men are more likely to brag out and/or exaggerate their sexual conquests, while women are more modest.)

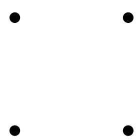
Some particularly useful graphs: Let G be a graph with vertices V and edges E, so we write $G = \{V, E\}$. If we select subsets $V' \subset V$ and $E' \subset E$, with the condition that each edge in E' joins vertices in V', then we get a **subgraph** $G' = \{V', E'\}$.

There are some particularly usefule graphs we'll mention. The **complete graph on n vertices**, which we write K_n has the vertex set $\{v_1, \dots, v_n\}$ and an edge $v_i - v_j$ so long as $i \neq j$, that is there's an edge joining all pairs of distinct vertices. Here's a pitcure of the complete graph on four vertices K_4:



K_4, the complete graph on four vertices

There's also the **empty graph** with n vertices, E_n. This graph has n vertices $\{v_1, \dots, v_n\}$ and no edges. Here's E_4:



E_4, the empty graph with four vertices

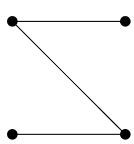
A **path** has the form $G = \{V, E\}$, where (after possibly renumbering the vertices) $V = \{v_1, \dots, v_n\}$ and the edges are

$$E = \{v_1 - v_2, v_2 - v_3, \dots, v_{n-1} - v_n\}.$$

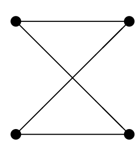
Similarly, a **cycle** has the form $V = \{v_1, \dots, v_n\}$ and

$$E = \{v_1 - v_2, v_2 - v_3, \dots, v_{n-1} - v_n, v_n - v_1\}.$$

In each case we observe that the vertices are all distinct, that is $v_i \neq v_j$ for $i \neq j$. Here are a path and a cycle with four vertices:



path



cycle

Exercise: List all the paths in K_4 starting at v_1 and ending at v_4.

Exercise: Show that a path with n vertices always has exactly $n-1$ edges, and that a cycle with n vertices always has exactly n edges.

We say a graph is **connected** if it contains a path connecting any two of its edges. A **tree** is a connected graph which doesn't contain any cycles. A vertex in a tree which has degree 1 is called a **leaf**.

Exercise: Show that the complete graph on n vertices is connected. Is it a tree?

Theorem 1. *Let $G = \{V, E\}$ be a tree. Then*

1. *There is a unique path between any pair of vertices.*
2. *Deleting any edge disconnects G.*
3. *Adding any edge creates a cycle.*
4. *Every tree with at least two vertices has at least two leaves.*
5. *$\#V = \#E + 1$.*

Proof. First we choose two vertices v_1 and v_2. The fact that G is connected means there is some path joining v_1 to v_2. Now suppose there are two different paths P_1 and P_2, both starting at v_1. Traverse along these paths simultaneously, until you get to the vertex u_1 where P_1 follows the edge $u_1 - u_2$ and P_2 follows the edge $u_1 - u_2'$. If we keep going along the path P_1 until we get to a vertex u_3 which also lies in P_2, we get a piece of the path P_1, which we call P_1'. Similarly, we denote by P_2' the piece of the path P_2 going from u_1 to u_3. Thus we've found a cycle by going along P_1' followed by P_2' backwards, which contradicts the fact that G is a tree.

Now suppose we delete an edge $e = v_1 - v_2$. This edge e was the unique path connecting v_1 to v_2, so we've just disconnected G.

Choose two vertices v_1 and v_2 which are not connected by an edge. Then there's a path P joining v_1 to v_2, so adding this edge $v_1 - v2$ creates a cycle.

Let G be a tree with at least two vertices, and let P be a path such that there are no other paths in G with more edges. Traversing along P we see the vertices $v_1, v_2, \ldots, v_m$ (listed in this order). If v_1 and v_m were not both leaves, then we could extend P to find a longer path, which contradicts the definiton of P. So we've found two leaves in G.

Finally, we show $\#V = \#E + 1$. (This is an induction argument, which is very common in graph theory.) If G only has one vertex, then it can't have any edges, so $\#V = 1$ and $\#E = 0$ and the formula holds. Now suppose a tree with n vertices has $n-1$ edges, and let G be a tree with $n+1$ vertices. Let v be a leaf of G. If we delete v and the one edge incident to it, we get a subtree $G' = \{V', E'\}$. Here $\#E' = \#E - 1$ and $\#V' = \#V - 1 = n$, so $\#V' = \#E' + 1$. The formula $\#V = \#E + 1$ follows. □

Trees appear in many areas of science, from biology (animal classifications) to computer science(searching algorithms). In fact, any connected $G = \{V, E\}$ contains what is called a **spanning tree**, that is a tree $G' = \{V, E'\}$ with the same vertex set.

Walks and tours: Here we describe various ways to traverse a graph. A **walk** on a graph G is an alternating list of vertices and edges of the form

$$v_0, v_0 - v_1, v_1, v_1 - v_2, \ldots, v_{n-1} - v_n, v_n.$$

This is pretty similar to a path, and in fact any path is a walk. However, a walk can include a cycle, and it can go over an edge or a vertex many times. A walk is **closed** if $v_0 = v_n$, that is if it starts and ends at the same place.

Exercise: Show that you can shorten any walk to a path that has the same starting and ending vertices as the original walk.

An **Euler walk** on a graph G is a walk which goes over every edge of G exactly once, and an **Euler tour** is an Euler walk which is closed. These are named after Leonard Euler, who invented graph theory in

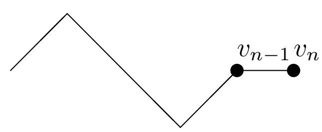
Theorem 2. *A connected graph has a Euler tour if and only if every vertex has even degree.*

Proof. Suppose a graph $G = \{V, E\}$ has an Euler tour. If we see a vertex v as we traverse along this walk k times, we arrive at v along k distinct edges, and leave v along k more distinct edges. Thus $\deg(v) = 2k$, and so every vertex has even degree.

Now suppose every vertex of G has even degree, and let W be the longest walk in G which traverses each edge of G at most once. We write W as

$$W = v_0, v_0 - v_1, v_1, v_1 - v_2, v_2, \dots, v_{n-1}, v_{n-1} - v_n, v_n.$$

Now, if $v_0 \neq v_n$ and we can't extend this walk, then $v_{n-1} - v_n$ must be the only edge incident to v_n, which means $\deg(v_n) = 1$, contradicting the fact that every vertex has even degree. (See the picture below.) So we must have $v_0 = v_n$.



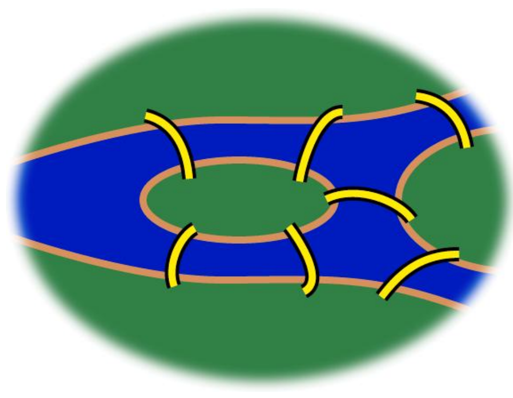
Finally we show W is an Euler tour. Suppose otherwise, then there is some other edge in G not in W. However, G is connected, so this edge must be incident to some vertex v_i in W. Write this mysterious edge as $u - v_i$. Now, u is not a vertex in W, so we can construct the walk

$$W' = u, u - v_i, v_i, v_i - v_{i+1}, dots, v_{n-1} - v_n, v_n, v_0 - v_1, v_1, v_1 - v_2, \dots, v_{i-1} - v_i, v_i.$$

Here we've used the fact that $v_0 = v_n$, as we've shown. This new walk W' is longer than W, which is a contradiction. Therefore, the walk W must traverse each edge exactly once, and so it's an Euler tour. $\square$

Corollary 3. *A connected graph has an Euler walk if and only if the number of vertices with odd degree is either* 0 *or* 2.

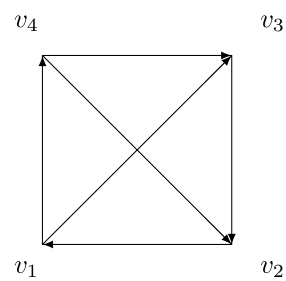
Exercise: Resolve Euler's seven bridges of Königsberg problem: can you walk over each of the seven bridges of Königsberg exactly once? See the picture below:



A **Hamiltonian walk** is a walk that visits every vertex of a graph exactly once, and a **Hamiltonian cycle** is a walk that starts and ends at the same vertex, and traverses every other vertex exactly once. In general, it is very difficult to determine whether a graph has a Hamiltonian walk or a Hamiltonian cycle.

Hamitonian walks can be useful for ordering data. For instance, suppose we have a round robin rugby tournament with n teams. This means every team plays each of the other $n-1$ teams exactly once. Ok, so after this tournament, how do you determine a winner? How would you rank the teams. If there are only three teams, this can already be difficult: team A could beat team B, then team B could beat team C, then team C could beat team A.

We can order these teams if we write out a directed graph and find a directed walk; let's do this. First we need a directed graph $G = \{V, E\}$ We write a vertex for each of the n teams, so we have vertices $v_1, v_2, \ldots, v_n$. Now we drawn an edge *from* v_i *to* v_j if v_i beat v_j. Notice that if we forget these direction arrows we just have the complete graph on n vertices, but the directions are important! They tell us who beat whom. Here's a picture of what could happen with four teams:

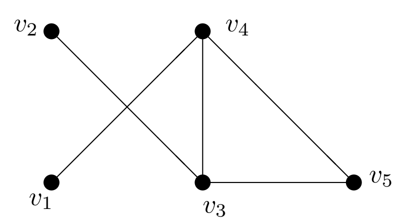


In this case, we see that v_1 beat v_3 and v_4, but lost to v_2; that v_3 beat v_2 but lost to v_1 and v_4; and so on. So who's the best team? Well, one way to rank everyone is to draw a Hamiltonian walk which follows the directions of all the arrows. If this walk looks like

$$W = v_1, v_1 \to v_2, \ldots, v_{n-1} \to v_n, v_n,$$

then we have our ranking: v_1, which beat v_2, which beat v_3, etc. It is a fact that we can always find a directed Hamiltonian walk for these round robin tournament graphs. Try to prove it!

Adjaceny matrices: It might seem like it's difficult to store the information of a graph in a computer, but actually it's easy. The key tool is something called the **adjacency matrix** of a graph. If $G = \{V, E\}$ is a graph with n vertices, then its adjacency matrix A is an $n \times n$ matrix, all of whose entries are 0 or 1. Write the vertices of G as $V = \{v_1, v_2, \ldots, v_n\}$. Then the entry A_{ij} is 1 if there's an edge joining v_i to v_j, and $A_{ij} = 0$ otherwise. For instance, here's a graph:



and here's its adjacency matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Exercise: Write down the adjacency matrix for the complete graph on four vertices. Then write down the adjacency matrix for the tournament graph with four teams which we drew above.

Exercise: Show that for an undirected graph an adjacency matrix always satisfies $A_{ij} = A_{ji}$. Is this still true for a directed graph?

Theorem 4. *Let G be a directed graph with vertices $v_1, v_2, \ldots, v_n$, and let A be its adjacency matrix. Then the entry of A^k in the ith row, jth column, that is $(A^k)_{ij}$, is exactly the number of directed walks from v_i to v_j of length k.*

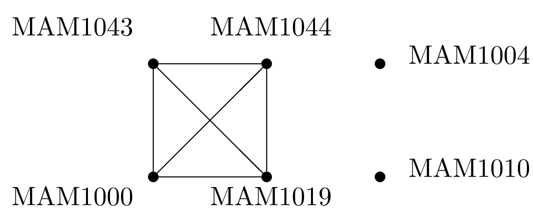
Proof. If $k = 1$ then the number of directed walks from v_i to v_j of length 1 is either 0 or 1. This number is 1 if there's an edge from v_i to v_j, and 0 otherwise. By definition, this is the corresponding entry of A.

Now we suppose for some k that the number of directed walks from v_i to v_j is exactly $(A^k)_{ij}$, and let's count the number of length $k+1$ walks from v_i to v_j. We can break up any length $k+1$ walk from v_i to v_j as a length k walk from v_i to some intermediate vertex v_m followed by a length 1 walk from v_m to v_j. (This second part can only be the edge from v_m to v_j.) Now, we already know that the number of length k walks from v_i to v_m is $(A^k)_{im}$, and that $A_{mj} = 1$ if there's a directed edge $v_m \to v_j$ and $A_{mj} = 0$ otherwise. To count up all the possible length $k+1$ walks from v_i to v_j, we sum over all the possible vertices v_m to get

$$\#\text{ length } k{+}1 \text{ walks from } v_i \text{ to } v_j = (A^k)_{i1}A_{1j} + (A^k)_{i2}A_{2j} + \cdots + (A^k)_{in}A_{nj} = (A^k \cdot A)_{ij} = (A^{k+1})_{ij}.$$

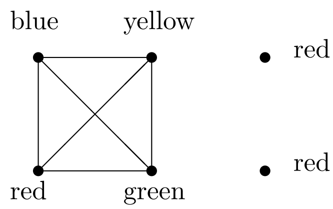
$\square$

Coloring graphs: Let's suppose we've lost faith in the exams office's ability to schedule exams, and so we want to make the exam schedule for all UCT students. What sort of information do we need to do this? Well, we need a list of all the courses which have exams, and then we need a list of which of these courses have students in common, and so on. We can organize this information as a graph! We draw a vertex for each course and draw an edge between two vertices if they have students in common. For instance, the part of the graph for (some of) the first year mathematics courses might look like this:



Hm, did that make the answer any easier? How many different exam periods do we need? Well, we can see that we need three different exam periods, because MAM1000, MAM1019, MAM1043, and MAM1044 all share students. If we look at the graph, we also see that the maximum degree of any vertex is three. Is this an accident? No! To make sense of this, we need another definition.

For a positive integer k, a graph G is called **k-colorable** if one can assign one of k colors to each vertex of G in such a way that no two adjacent vertices have the same color. The minimum k such that G is k-colorable is called the **chromatic number** of G. For instance, here's a coloring of the graph we made above with 4 colors:



Exercise: Can you find a coloring of this graph with 3 colors? Or does it have chromatic number 4?

Theorem 5. *If every vertex of G has degree at most k then G is $(k+1)$-colorable.*

Proof. If G has 1 vertex, then it can't have any edges, and you can color it with one color. So the statement holds for the graph with one vertex.

Now suppose that every graph with n vertices such that each vertex has degree at most k is $(k+1)$-colorable, and let G be a graph with $n+1$ vertices, such that each vertex also has degree at most k. Now remove any vertex v from G, it doesn't matter which one. You're left with a graph G' which has n vertices and each vertex has degree at most k, so it's $(k+1)$-colorable. Now add v back in. It connects to G' with at most k edges, so we have at least 1 color left over to choose for the color of v, which means G is $(k+1)$-colorable. Now we've shown that all graphs such that each vertex has degree at most k is $(k+1)$-colorable. $\square$

The most famous application of coloring graphs (and the origin of the name) is the problem of coloring countries on a map. Here's a map of Africa:

Can you color all the countries on this map using six colors such that no two countries with the same color share a border? How about five colors? How many colors do you think you really need?

The case of 2-colorable graphs are so special they have a special name: **bipartite graphs**. The graph of men and women we saw all the way back at the beginning of these notes is an example of a bipartit graph. For instance, you could color all the male vertices blue and all the red vertices red.

Exercise: Show that every tree is a bipartite graph.

Exercise: Show that every cycle with an even number of vertices is bipartite.

In fact, there's a nice characterization of bipartite graphs.

Theorem 6. *A graph is bipartite if and only if it does not contain a cycle with an odd number of vertices.*

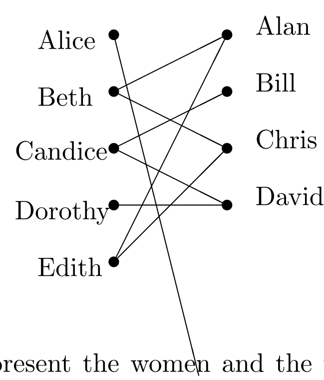
Proof. Suppose $G = \{V, E\}$ has a cycle with an odd number of vertices, which we write as

$$C = \{v_1, v_1 - v_2, v_2, \ldots, v_{2k} - v_{2k+1}, v_{2k+1}, v_{2k+1} - v_1\},$$

and attempt to color the graph with the two colors blue and green. The choice of which two colors we use is unimportant, you can use two other colors if you don't like blue or green. We pick one color, say blue, for v_1, but then the next vertex on our list for the first vertex v_1. Then v_2 has to be the other color, say green. Now we've exhausted our choice of colors, and the rest of the vertices in the cycle C must alternate blue and green, with the even number vertices being green and the odd numbered vertices being blue. But then we get to the end of this list and see that the blue vertex v_{2k+1} is adjacent to the blue vertex v_1, which means we need more than colors for the cycle C. Therefore we the whole graph G also needs more than two colors.

Notice that we've just proven that no cycle with an odd number of vertices is 2-colorable. Now, if G is bipartite, *i.e.* 2-colorable, then this coloring will restrict to any cycle C in G, which implies that G cannot have any cycle with an odd number of vertices. $\square$

Bipartite graphs are very useful for matching problems. To illustrate this, we consider a group of men and women, and suppose we want to match them into marriagable pairs. Of course, we should match every woman with a man she likes, otherwise the marriage is doomed. To model this problem, we draw a graph representing all the men and women in our group, and which women like which men. This is very much like the graph we drew for the social sciences study we considered at the beginning of these notes, and part of it might look like this:



Here, the vertices on the left represent the women and the vertices on the right represent the men. Given a woman vertex w and a man vertex m, there is an edge $w - m$ precisely if w likes m.

We want to find a **marriage matching**. This would be a subset of the edges such that each woman has at least one edge incident, and each man has at most one edge incident to it. Such a matching would find a marriage partner for each woman, such that no man gets partnered with two women. (Notice that we do not require that each man gets a marriage partner.)

Observe that this graph is bipartite. In fact, we can color all the women one color (say, blue), and the men another color (say, green). This is because there are no edges between two women or two men.

Now, we can do at least part of the matching by hand. For instance, it might look like we can match Candice with either Bill or David, but actually we can't because Dorothy is picky on only likes David, which means we'd have to match Candice with Bill. Constructing the entire matching in this way would be very tedious, so we won't do it. Rather, before we start all that work, whether such a matching is even possible.

There's an obvious criterion we need in order that a matching exist: if we choose any subset of women, the set of men they like collectively must be at least as large. For instance, if there's a set of four women who (as a group) like only three men, then we can't match up these three women, so we can't match up the entire group. However, if every set of four women collectively likes at least four men then we stand a chance of finding a matching. This turns out to be a sufficient criterion.

Theorem 7. *Let $G = \{V, E\}$ be a women-men graph as above. That is, the vertex set is $V = W \cup M$, and the only edges join a woman vertex w to a man vertex m. For each subset $S \subset W$ we denote by $N(S)$ the set of all $m \in M$ such that there is an edge from some $w \in S$ to m. Then there is a marriage matching as described above if and only if $\#N(S) \geq \#S$ for all subsets $S \subset W$.*

Proof. First suppose that a marriage matching exists and choose a set S of women. Each woman in this set S gets married to the man determined by this matching, so $\#N(S) \geq \#S$ for this subset S.

Now suppose $\#N(S) \geq \#S$ for every subset S. We'll prove the theorem using induction on the number of women (*i.e.* the number of vertices in W). If $\#W = 1$, then there's at least one edge incident to that one vertex. Choosing one of these edges will give us a marriage matching.

We complete the proof using strong induction: assume the theorem is true for every woman-man graph with less than n vertices, and prove it's true for woman-man graphs with n vertices. If $\#W = n \geq 2$ we have the two possibilities listed below.

- We could have that every strictly smaller subset W' of women likes a strictly larger set of men. Symbolically, we can write this case as

  $$\#W' < \#W \Rightarrow \#N(W') > \#W'.$$

  In this case, choose an arbitrary woman, match her with one of the men she likes, and proceed. Now we have another woman-man graph with all the same conditions, but with one fewer woman. So we can work our way down in this fashion to pair off all the woman, and then we're done.

- It could happen that there's some smaller subset $W' \subset W$ such that $\#N(W') = \#W'$. Now, by the induction hypothesis, we can match the women in W' exactly with the men in $N(W') = M'$, and so we do that. After this pairing, we're left with the graph G'' whose vertex set is $V = W'' \cup M''$, where W'' is the set of women not in W', and M' is the set of men not in M''. If we show that this new woman-man graph satisfies the marriage matching condition, we're done. So let $S \subset W''$ and consider $N''(S)$, the set of men in M'' which the women in S likes. We want to show that $\#N'(S) \geq \#S$. However, we know that the combined set of women in $S \cup W'$ like the combined set of men in $N'(S) \cup N(W')$, which, because the original graph G satisfies the marriage matching condition, means

  $$\#(S \cup W') \leq \#(N'(S) \cup N(W')) \leq \#(N'(S)) + \#N(W').$$

  By pairing off all the women in W', we've subtracted $\#W' = \#N(W')$ from both sides of the inequality above, which leaves us with $\#S \leq \#N'(S)$, which means exactly that the new graph G' satisfies the marriage matching condition.

In both cases we've shown that if all woman-man graphs satisfying the marriage matching condition with fewer than n women have marriage pairings, then so do all such graphs with n women. Combined with the base case when the number of women is 1, we've completed the proof of our theorem. $\square$

We can restate this theorem in terms of general bipartite graphs, so that we don't always have to discuss finding marriage partners. If we have a graph $G = \{V, E\}$ and consider a subset $S \subset V$ of vertices, we denote its set of neighbors as $N(S)$. That is, $u \in N(S)$ precisely when there is a $v \in S$ adjacent to u.

Theorem 8. *Let $G = \{V, E\}$ be a bipartite graph, and write its vertex set as $V = L \cup R$ (corresponding to the two colors). Then there is a matching for the vertices in L, which associates exactly one vertex in R to each vertex in L, if and only if $\#N(S) \geq \#S$ for every subset $S \subset V$.*

This theorem is called Hall's theorem. It is a special case of the max-flow, min-cut theorem, and is a very useful tool in linear programming.