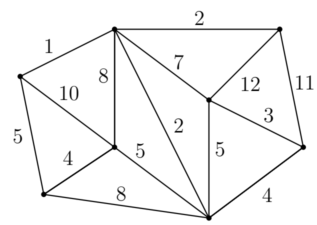
1 Minimum Spanning Trees

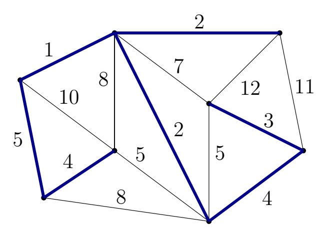
Given a graph $G = (V, E)$, remember that a spanning tree is a tree that "spans" all the nodes. In other words, it is a tree (connected acyclic graph) on all the nodes V.

Evert connected graph has a spanning tree. Some graphs may have several spanning trees. In fact, you already saw Cayley's formula, which says that the complete graph K_n on n vertices has n^{n-2} spanning trees.

Now suppose we are given a graph $G = (V, E)$, where the edges have costs. I.e., each edge $e \in E$ has a cost $c_e \in \mathbb{R}$. For example, here is such a graph (which happens to have positive integer costs).



And we want to find the spanning tree with the *least cost*, where the cost of the spanning tree $T = (V, E')$ is $\sum_{e \in E'} c_e$, the sum of its edge costs. Here is the minimum-cst spanning tree for the graph above.



Very often the minimum-cost spanning tree gets shortened to "minimum spanning tree" or "MST". There are many algorithms to solve this problem, you will probably see some of them in 15-451. Here is perhaps the simplest algorithm of them all.

1.1 Kruskal's Algorithm

This "greedy" algorithm for MST is due to Joseph Kruskal:

> Start off with an edge set $F = \emptyset$.
>
> Create a list L of the edges in non-decreasing order of weight.
>
> While L is not empty: pick a least-cost edge e from L.
>
> If adding e to the set F creates no cycles (i.e., it connects two components), set $F \leftarrow F \cup \{e\}$. OTOH, if adding e to F creates a cycle, just discard e.
>
> Return the subgraph $T = (V, F)$.

For example, you should check that running it on the example above gives the MST we claimed. Note that if we start off with a connected graph, we would definitely end up with a spanning tree. (Why?) The claim is that this tree has the least cost possible, among all spanning trees.

Theorem 1.1 *Kruskal's algorithm returns a spanning tree $T = (V, F)$ of minimum cost.*

We give the proof for the special case where all the edge costs are different. The proof easily extends to the general case.

Proof: Suppose not. Suppose there is a spanning tree $M = (V, E')$, such that its cost $\sum_{e \in E'} c_e < \sum_{e \in F} c_e$. Then there must be some step in the algorithm where we pick an edge e into F, which did not belong to E'.

Now consider adding e to M. Since M was a tree, adding e to it creates a single cycle C. Since T was acyclic, there must be some edge $f \in C$ that does not belong to T. So drop this edge, to get the new spanning tree $N = (M \cup \{e\}) \setminus \{f\}$. If $c_f > c_e$, then N would have a lower cost than M, and we would get a contradiction.

So $c_f < c_e$. (They cannot be equal, since we assumed distinct costs.) So f must have been considered before e by Kruskal's algorithm. But f does not belong to T, so the algorithm must have discarded it. Why? This would only happen if adding f to the current edge set F would create a cycle.

Hmm. All the edges in F at that time also belong to M (since all this happened before we reached e, and Kruskal agreed with the MST M until that time). And the edge f also belongs to M. So M contains a cycle. Which is a contradiction. ■

> **Exercise:** Extend the proof to the case where different edges may have the same costs. (Hint: where was the only point you used the assumption of distinct costs?)

1.2 An Application: Traveling Salesman Problem

Consider a setting where there are n cities, and a traveling salesman wants to find a tour of these cities. This is a permutation of the **To be completed.**

2 Bipartite Graphs

One type of graph which arises often in applications is a *bipartite graph.* In such a graph, the vertex set V can be partitioned into two parts A and $B = V \setminus A$, and all edges go between nodes in A and those in B. (I.e., $E \subseteq A \times B$.)

So, to show a graph is bipartite, you need to exhibit this partition of the vertex set—to emphasize this partition, we write a bipartite graph as a triple $G = (A, B, E)$ instead of just a tuple $G = (V, E)$. Also, we usually imagine the sets A as being on the left, and B on the right, and the edges going between the two sides—as in the following figure.

2.1 Bipartite Graphs and No Odd Cycles

Theorem 2.1 *A graph is bipartite if and only if it contains no cycles of odd length.*

Proof: Suppose $G = (A, B, E)$ is bipartite: then consider any cycle $C = v_1 - v_2 - v_3 - \ldots - v_k - v_1$ in G. As we traverse this cycle, we alternate between the sides: if v_1 is on the left, then all nodes v_j for j odd must be on the left, and v_j for even j on the right. Now there is an edge from v_k to v_1, so if v_1 is on the left v_k is on the right, and hence the length of the cycle k is even.

Conversely, suppose G has no cycles of odd length. (If G is not connected, do the following on each connected component of G.) Pick any vertex v and perform a breadth-first search from v. Recall that in BFS, all edges either go between adjacent levels, or within a level. But if there is some edge between two vertices on level ℓ (where the root is at level 0), this forms a cycle of length $2\ell + 1$, contradicting the fact that there are no odd cycles. So the only edges must go between levels. Define A to be the even-numbered levels of this BFS, and B to be the odd-numbered levels: then $G = (A, B, E)$ is a bipartite graph. ■

Since a tree does not have any cycles at all (and hence no odd cycles), a tree is bipartite.

2.2 Bipartite Graphs and Graph Colorings

We often study *colorings* of graphs: these are maps that assign colors to the vertices of the graph. We say a coloring is *proper* if no two vertices that are connected by an edge have the same color. In other words, the coloring does not color the endpoints of any edge the same. [1]

A graph G is k-colorable if there exists a proper coloring for G that uses at most k colors. Note that a graph is 1-colorable if and only if it does not have any edges at all. The next simplest case is already interesting —

Theorem 2.2 *A graph G is two-colorable if and only if it is bipartite.*

Proof: If $G = (V, E)$ is bipartite, then V can be partitioned into two parts A and $B = V \setminus A$ such that all edges go between A and B. Hence we can color all vertices in A the same color, and those in B the other color.

Conversely, suppose G is two-colorable. Let A be the vertices of color 1 and B be the remaining vertices. No edges can go between the nodes in A, or nodes in B, hence $G = (A, B, E)$ is a bipartite graph. ■

Combining Theorem 2.2 with Theorem 2.1 tells us that coloring any odd cycle requires at least 3 colors.

[1] The study of *edge-colorings* of graphs is also an interesting topic: here, we assign colors to the edges, such that no two edges incident to the same vertex have the same color. However, when we talk about colorings, it's usually vertex colorings

3 Matchings

Given a graph $G = (V, E)$, a *matching* in G is a subset $M \subseteq E$ of edges such that no two edges in M share a node. In other words, each node in V is incident to either zero or one edge in M.

> **Exercise:** How many matchings are there in the following graphs?

A matching M in the graph G is called a *perfect matching* if each vertex in V has exactly one edge incident to it.

> **Exercise:** Which of the following graphs have perfect matchings?

A matching M in the graph G is said to be a *maximum matching* if for all other matchings M' in G, $|M| \geq |M'|$. I.e., M is a matching with the largest number of edges possible.

> **Exercise:** What is the size of maximum matchings in the following graphs?

A matching M is said to be *maximal* if for all edges $e \in E \setminus M$, $M + e$ is not a matching. Note that a maximal matching need not be a maximum matching—for example:

Theorem 3.1 *For any graph, if M is a maximal matching, then for all matchings M', $|M| \geq |M'|/2$.*

Proof: Consider any edge $e' = \{u, v\}$ in the matching M'. We claim that the maximal matching M must contain some edge e' that is incident to u or to v. (It may well be the case that $e' = e$.) Indeed, if neither u nor v was matched in M, we could add in the edge e' to M, which would contradict the maximality of M.

OK, so call such an edge $e \in M$ the edge that "blocks" $e' \in M'$. (If there are two such edges, choose one of them.) Hence, each edge in M' is blocked by some edge in M. But each edge in M can only block two edges in M', one at each of its endpoints. Hence, $|M'| \leq 2|M|$. ■

And the example below shows that there are indeed maxi*mal* matchings which have only half as many edges as maxi*mum* matchings.

4 The Marriage Theorem

In this section, we will prove an interesting theorem about perfect matchings in bipartite graphs. It comes with it's own little story:

> At a dance, there are n girls and n boys. If for every set S of the girls, at least $|S|$ boys are liked by at least one girl in S, then we can pair the boys and girls so that each girl is paired with a boy she likes.

To make this a little more formal, we use some jargon from graph theory. Define the *neighbors of sets of nodes*: given a set $S \subseteq V$, the neighbor set $N(S)$ consists of all $u \in V$ which is connected by an edge to some vertex $v \in S$. Note that this is the same as $\cup_{v \in S} N(v)$.

Now if we think of the girls as forming a set A, the boys as the set B, and an edge between $a \in A$ and $b \in B$ meaning that a likes b, we get the following statement.

Theorem 4.1 (Hall's Marriage Theorem) *For any bipartite graph $G = (A, B, E)$, G has a perfect matching if and only if the following two conditions are true:*

- $|A| = |B|$, *and*
- *for every subset $S \subseteq A$, $|N(S)| \geq |S|$.*

Before we prove this theorem, let us look at a few examples of bipartite graphs, and check that we can either find a perfect matching, or an "offending" set S such that $|N(S)| < |S|$.

One more thing to observe. The second "expansion" condition says that each subset S of the left hand side A has at least $|S|$ neighbors. Can it be the case that both conditions hold, yet some subset S' of the right hand side B has fewer than $|S'|$ neighbors? No, in fact, and here's the simple proof!

Claim 4.2 (Symmetry Lemma) *For a bipartite graph $G = (A, B, E)$ with $|A| = |B|$, if $|N(S)| \geq |S|$ for every subset $S \subseteq A$, then $|N(S')| \geq |S'|$ for every subset $S' \subseteq B$.*

Proof: For a contradiction, suppose there is some such $S' \subseteq B$ with $|N(S')| < |S'|$. Then consider the set $S = A \setminus N(S')$. All edges from S must go to $B \setminus S'$, since none of the nodes in S are neighbors of any nodes in S', so $N(S) \subseteq B \setminus S'$. Hence,

$$|N(S)| \subseteq |B \setminus S'| = |B| - |S'| < |A| - |N(S')|,$$

since $|A| = |B|$ and $|N(S')| < |S'|$. But $A \setminus N(S') = S$, and hence $|A| - |N(S')| = |S|$. Substituting, we get $|N(S)| < |S|$, a contradiction. ■

Let's define a bipartite graph $G = (A, B, E)$ to be *matchable* if

- (Size condition) $|A| = |B|$, and
- (Expansion condition) *either* for every subset $S \subseteq A$, $|N(S)| \geq |S|$, or for every subset $S \subseteq B$, $|N(S)| \geq |B|$.

By the symmetry lemma, if one part of the expansion condition is satisfied, then the other part is also satisfied.

4.1 Proof of the Marriage Theorem

Now back to the proof of the marriage theorem. By the Symmetry Lemma, the Marriage theorem is equivalent to showing that a bipartite graph has a perfect matching if and only if it is matchable. And this is what we will show.

The proof proceeds by trying to just match some pair arbitrarily and hoping that does not mess up things. Either it doesn't (so we're done by induction), else the fact that we messed up allows us to split the graph into two smaller bipartite graphs and induct on each of them!

Proof: Let us prove the easier direction first: If G has a perfect matching M, then G is matchable. Since the perfect matching $M \subseteq E \subseteq A \times B$, each vertex in A is matched to some vertex in B and vice versa. Hence $|A| = |B|$. Moreover, at least $|S|$ of the edges in M go from S into $N(S) \subseteq B$ (since the graph is bipartite and M is a perfect matching), and each of these edges must go to a different vertex in $N(S)$ (since M is a matching). So $|N(S)| \geq |S|$.

Now for the other direction: *if the graph is matchable, it must have a perfect matching.* It's just a little longer, but fairly easy as well. We know that $|A| = |B|$, and that $|N(S)| \geq |S|$ for all $S \subseteq A$ (and also for all $S \subseteq B$, by Claim 4.2).

Let's start off by picking some $a \in A$ and match it to some $b \in B$ such that a, b are connected by an edge. Now remove a, b and all their edges from G to get $G_2 = (A', B', E')$. Clearly $|A'| = |B'|$ (we removed one vertex from each of A and B), so G_2 is matchable then we're done by induction.

So the bad case is when G_2 is not matchable, and some $S \subseteq A'$ has fewer than $|S|$ neighbors in B'. Since we only removed b, the size of S's neighborhood in G, namely $|N(S)|$, must have been *equal to* $|S|$, and removing b has caused its neighborhood to be deficient in the new graph H. So now construct two new graphs, H_1 with node sets S and $N(S)$ and all the edges between them, and H_2 with all the remaining nodes in G and the edges between them.

Note that b was in $N(S)$ and hence is now in H_1, and $a \notin S$ and hence now is in H_2. So both H_1 and H_2 are non-empty, and smaller than G. If we show that both graphs H_1 and H_2 were matchable, we would be done by induction.

- Size condition: by our construction, $|S| = |N(S)|$, and so both H_1 and H_2 have equal number of nodes on the left and the right.

- Expansion condition: note the only edges we have removed are between $A \setminus S$ and $N(S)$. (E.g., (a, b) is such an edge.)

  Since every subset T of S had at least $|T|$ neighbors in G, and no edges incident to S (which is the left vertex set of H_1) are removed, the expansion condition is satisfied for H_1.

  Sinilarly, no edges from the right hand side of H_2 are removed, so each set T on the right in H_2 continues to have the same number of neighbors on the left as in G. So the expansion condition is satisfied for H_2.

Hence both H_1 and H_2 are matchable, and we are done! ■

4.2 Some Corollaries of the Marriage Theorem

A simple corollary of the marriage theorem is the following:

Theorem 4.3 *A bipartite graph $G = (A, B, E)$ where each node in G has exactly $k \geq 1$ edges incident on it, always contains a perfect matching.*

Note that the case $k = 1$ is only possible if the graph itself just consists of a matching and nothing else. What about higher k?

Proof: To show this, it suffices to show that G satisfies the conditions of the Marriage theorem. If each node has exactly k edges, and all edges go from A to B, then $|E| = k|A| = k|B|$. So $|A| = |B|$.

Now for the second condition. For any set $S \subseteq A$, there are $k|S|$ edges leaving S. They must go to $N(S)$, so the number of edges reaching $N(S)$ is at least $k|S|$ (as $N(S)$ might have even some edges going to $A \setminus S$). But each node in $N(S)$ has only k edges incident to it, so $|N(S)| \geq |S|$. Done! ■

And here are two problems that can be easily solved using Hall's theorem:

Problem: *The n people of Clubtown have formed n clubs. Each club has as members some subset of the population, and each person in town potentially belongs to several clubs. However, the clubs have the property that the union of every k of these clubs has size at least k. Show that you can choose one leader from each club, so that no person is leader for more than one club.*

> Construct a bipartite graph with the n nodes on the left each representing a club, and the nodes on the right representing people. There is an edge between a club and a person if the person belongs to the club. This satisfies the conditions of Hall's theorem, since the union of any subset of k clubs on the left contains at least k people. Hence, there exists a perfect matching between clubs and people: choose the leader of a club to be the person matched to it.

Problem: *Suppose that a standard deck of cards is dealt into 13 piles of 4 cards each. Then it is possible to select a card from each pile so that the 13 chosen cards contain exactly one card of each rank.*

Construct a bipartite graph with the 13 nodes on the left each representing a rank, and the nodes on the right representing the 13 plies. There is an edge between a rank and a pile if the pile contains a card of that rank—hence each edge corresponds to a card.

Each vertex on the left has degree 4 (since there are 4 cards of that rank), and so does each vertex on the right (since each pile has 4 cards). So we can just apply Theorem 4.3 to infer there is a perfect matching from the ranks to the piles. Now choose the cards corresponding to these matching edges: there's one card for each rank, and one card from each pile.

5 To Do

- add more exercises! add more figures!
- add in MST notes.