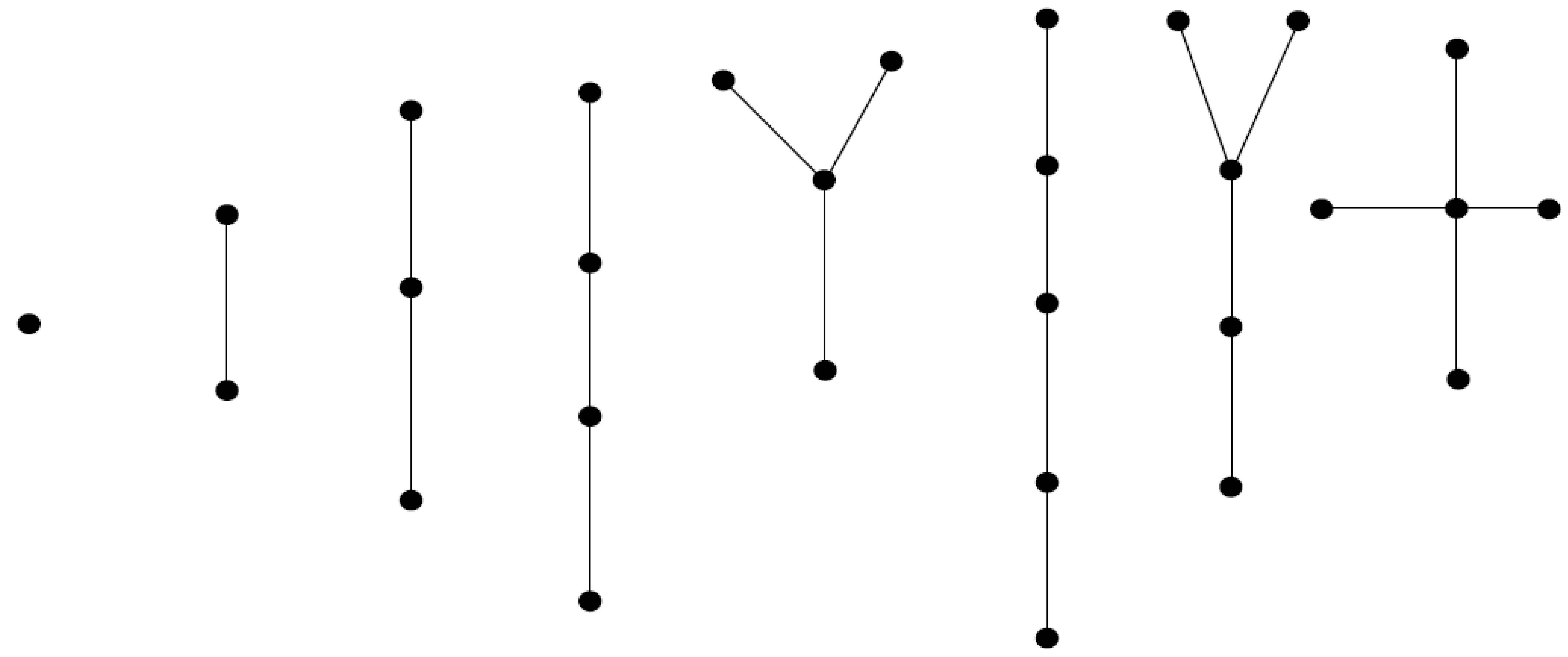
Trees and Forest

Definition: A graph having no cycles is said to be *acyclic*. A *forest* is an acyclic graph.

Definition: A *tree* is a connected graph without any cycles, or a tree is a connected acyclic graph. The edges of a tree are called *branches*. It follows immediately from the definition that a tree has to be a simple graph (because self-loops and parallel edges both form cycles). Figure 4.1(a) displays all trees with fewer than six vertices.



(a)

Fig. 4.1(a)

The following result characterises trees.

Theorem 4.1 A graph is a tree if and only if there is exactly one path between every pair of its vertices.

Proof Let G be a graph and let there be exactly one path between every pair of vertices in G. So G is connected. Now G has no cycles, because if G contains a cycle, say between vertices u and v, then there are two distinct paths between u and v, which is a contradiction. Thus G is connected and is without cycles, therefore it is a tree.

Conversely, let G be a tree. Since G is connected, there is at least one path between every pair of vertices in G. Let there be two distinct paths between two vertices u and v of G. The union of these two paths contains a cycle which contradicts the fact that G is a tree. Hence there is exactly one path between every pair of vertices of a tree. ❑

The next two results give alternative methods for defining trees.

Theorem 4.2 A tree with n vertices has $n-1$ edges.

Proof We prove the result by using induction on n, the number of vertices. The result is obviously true for $n = 1$, 2 and 3. Let the result be true for all trees with fewer than n vertices. Let T be a tree with n vertices and let e be an edge with end vertices u and v. So the only path between u and v is e. Therefore deletion of e from T disconnects T. Now, $T-e$ consists of exactly two components T_1 and T_2 say, and as there were no cycles to begin with, each component is a tree. Let n_1 and n_2 be the number of vertices in T_1 and T_2 respectively, so that $n_1+n_2 = n$. Also, $n_1 < n$ and $n_2 < n$. Thus, by induction hypothesis, number of edges in T_1 and T_2 are respectively n_1-1 and n_2-1. Hence the number of edges in $T = n_1-1+n_2-1+1 = n_1+n_2-1 = n-1$. ❑

Theorem 4.3 Any connected graph with n vertices and $n-1$ edges is a tree.

Proof Let G be a connected graph with n vertices and $n-1$ edges. We show that G contains no cycles. Assume to the contrary that G contains cycles.

Remove an edge from a cycle so that the resulting graph is again connected. Continue this process of removing one edge from one cycle at a time till the resulting graph H is a tree. As H has n vertices, so number of edges in H is $n-1$. Now, the number of edges in G is greater than the number of edges in H. So $n-1 > n-1$, which is not possible. Hence, G has no cycles and therefore is a tree. ❑

The following results give some more properties of trees.

Theorem 4.5 A graph G with n vertices, $n-1$ edges and no cycles is connected.

Proof Let G be a graph without cycles with n vertices and $n-1$ edges. We have to prove that G is connected. Assume that G is disconnected. So G consists of two or more components and each component is also without cycles. We assume without loss of generality that G has two components, say G_1 and G_2 (Fig. 4.1(b)). Add an edge e between a vertex u in G_1 and a vertex v in G_2. Since there is no path between u and v in G, adding e did not create a cycle. Thus $G \cup e$ is a connected graph (tree) of n vertices, having n edges and no cycles. This contradicts the fact that a tree with n vertices has $n-1$ edges. Hence G is connected.

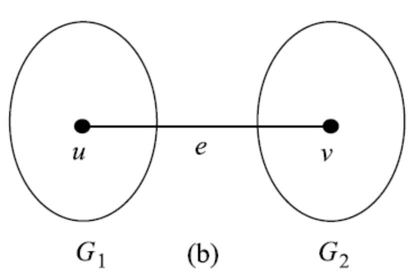


Fig. 4.1(b)

Theorem 4.6 Any tree with at least two vertices has at least two pendant vertices.

Proof Let the number of vertices in a given tree T be $n (n > 1)$. So the number of edges in T is $n-1$. Therefore the degree sum of the tree is $2(n-1)$. This degree sum is to be divided among the n vertices. Since a tree is connected it cannot have a vertex of 0 degree. Each vertex contributes at least 1 to the above sum. Thus there must be at least two vertices of degree exactly 1.

Theorem 4.8 A forest of k trees which have a total of n vertices has $n-k$ edges.

Proof Let G be a forest and $T_1, T_2, \ldots, T_k$ be the k trees of G. Let G have n vertices and $T_1, T_2, \ldots, T_k$ have respectively $n_1, n_2, \ldots, n_k$ vertices. Then $n_1 + n_2 + \ldots + n_k = n$. Also, the number of edges in $T_1, T_2, \ldots, T_k$ are respectively $n_1 - 1, n_2 - 1, \ldots, n_k - 1$. Thus number of edges in $G = n_1 - 1 + n_2 - 1 + \ldots + n_k - 1 = n_1 + n_2 + \ldots + n_k - k = n - k$. ❑

4.3 Number of Labelled Trees

$a, b) \cap (b, a)$

Let us consider the problem of constructing all simple graphs with n vertices and m edges. There are $n(n-1)/2$ unordered pairs of vertices. If the vertices are distinguishable from each other (i.e., labelled graphs), then the number of ways of selecting m edges to form the graph is $\binom{\frac{n(n-1)}{2}}{m}$.

Thus the number of simple labelled graphs with n vertices and m edges is

$$\binom{\frac{n(n-1)}{2}}{m}. \tag{A}$$

Clearly, many of these graphs can be isomorphic (that is they are same except for the labels of their vertices). Thus the number of simple, unlabelled graphs of n vertices and m edges is much smaller than that given by (A) above.

Theorem 4.20 The number of simple, labelled graphs of n vertices is $2^{\frac{n(n-1)}{2}}$.

Proof The number of simple graphs of n vertices and 0, 1, 2, $\ldots$, $n(n-1)/2$ edges obtained by substituting 0, 1, 2, $\ldots$, $n(n-1)/2$ for m in (A). The sum of all such numt is the number of all simple graphs with n vertices.

Therefore the total number of simple, labelled graphs of n vertices is

$$\binom{\frac{n(n-1)}{2}}{0}+\binom{\frac{n(n-1)}{2}}{1}+\binom{\frac{n(n-1)}{2}}{2}+\ldots+\binom{\frac{n(n-1)}{2}}{\frac{n(n-1)}{2}}=2^{\frac{n(n-1)}{2}},$$

by using the identity $\binom{k}{0}+\binom{k}{1}+\binom{k}{2}+\ldots+\binom{k}{k}=2^k$.

Theorem 4.22 (Cayley) There are n^{n-2} labelled trees with n vertices, $n \geq 2$.

Proof Let T be a tree with n vertices and let the vertices be labelled $1, 2, \ldots, n$. Remove the pendant vertex (and the edge incident to it) having the smallest label, say u_1. Let v_1 be the vertex adjacent to u_1. From the remaining $n-1$ vertices, let u_2 be the pendant vertex with the smallest label and let v_2 be the vertex adjacent to u_2. We remove u_2 and the edge incident on it. We repeat this operation on the remaining $n-2$ vertices, then on $n-3$ vertices, and so on. This process completes after $n-2$ steps, when only two vertices are left.

Let the vertices after each removal have labels $v_1, v_2, \ldots, v_{n-2}$. Clearly, the tree T uniquely defines the sequence

$$(v_1, v_2, \ldots, v_{n-2}). \tag{4.22.1}$$

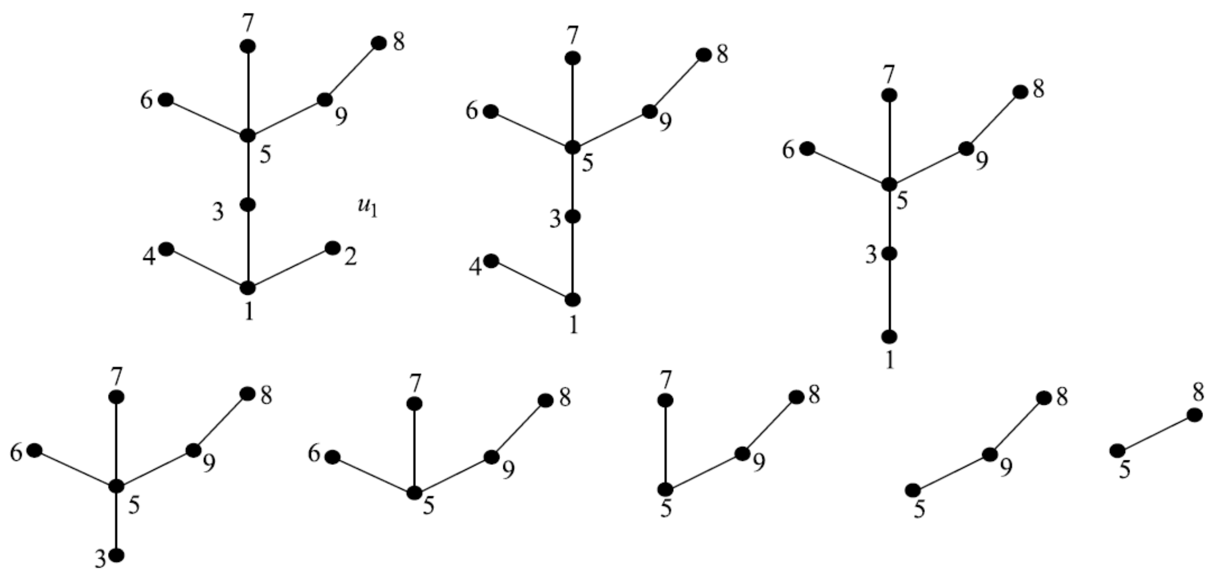
Conversely, given a sequence of $n-2$ labels, an n-vertex tree is constructed uniquely as follows. Determine the first number in the sequence

$$1, 2, 3, \ldots, n, \tag{4.22.2}$$

that does not appear in (4.22.1). Let this number be u_1. Thus the edge (u_1, v_1) is defined. Remove v_1 from sequence (4.22.1) and u_1 from (4.22.2). In the remaining sequence of (4.22.2), find the first number which does not appear in the remaining sequence of (4.22.1). Let this be u_2 and thus the edge (u_2, v_2) is defined. The construction is continued till the sequence (4.22.1) has no element left. Finally, the last two vertices remaining in (4.22.2) are joined.

For each of the $n-2$ elements in sequence (4.22.1), we choose any one of the n numbers, thus forming n^{n-2} $(n-2)$-tuples, each defining a distinct labelled tree of n vertices. Since each tree defines one of these sequences uniquely, there is a one–one correspondence between the trees and the n^{n-2} sequences. ❑

Example Consider the tree shown in Figure 4.11. Pendant vertex with smallest label is u_1. Remove u_1. Let v_1 be adjacent to u_1 (label of v_1 is 1). Pendant vertex with smallest label is 4. Remove 4. Here 4 is adjacent to 1. Pendant vertex with smallest label is 1. Remove 1. Here 1 is adjacent to 3. Remove 3. Then 3 is adjacent to 5. Remove 6. So 6 is adjacent to 5. Remove 5. Remove 7. 7 is adjacent to 5. So 5 is adjacent to 9. Sequence $(v_1, v_2, \ldots, v_{n-2})$ is (1, 1, 3, 5, 5, 5, 9).



Number of unlabeled trees

As described in the case of labeled trees, a similar approach can give an upper bound on number of unlabeled trees.

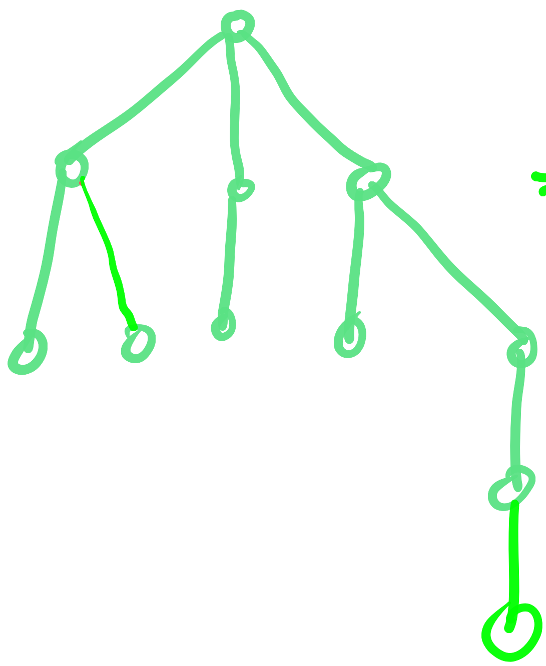
For a given unlabeled tree, a binary string can be constructed as follows

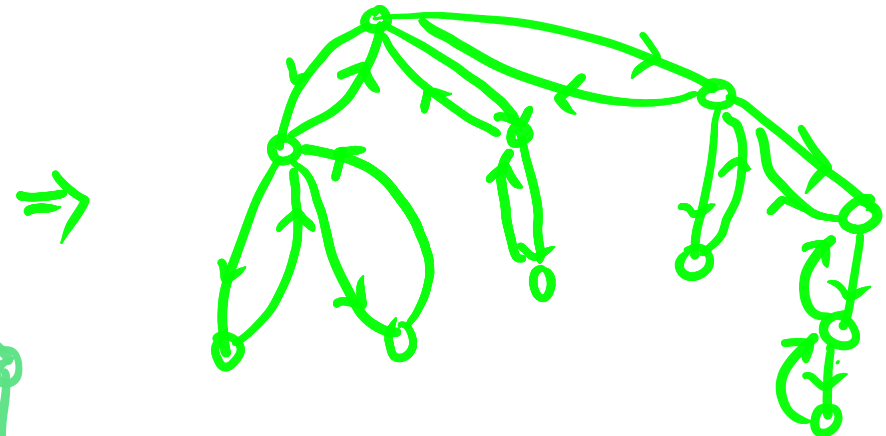
1. Pick a vertex as the root
2. Find an eulerian tour that visits every edge exactly twice.
3. Follow the eulerian tour and compute a string as follows

3.1 If an edge is visited downwards, add a 1 in the sequence.

3.2 If an edge is visited upward, add a 0 to the sequence.

Example





1101001100

11011110000

The length of the string is $2(n-1)$

Given such a binary encoding of a tree, the tree can be constructed.

The above strategy imply the following two results

(1) A tree can be stored using $2(n-1)$ bits

(2) There are atmost $2^{2(n-1)}$ trees.

Minimum Spanning Trees

Suppose we are given a connected, undirected, *weighted* graph. This is a graph $G = (V, E)$ together with a function $w: E \to \mathbb{R}$ that assigns a real weight $w(e)$ to each edge e, which may be positive, negative, or zero. This chapter describes several algorithms to find the ***minimum spanning tree*** of G, that is, the spanning tree T that minimizes the function

$$w(T) := \sum_{e \in T} w(e).$$

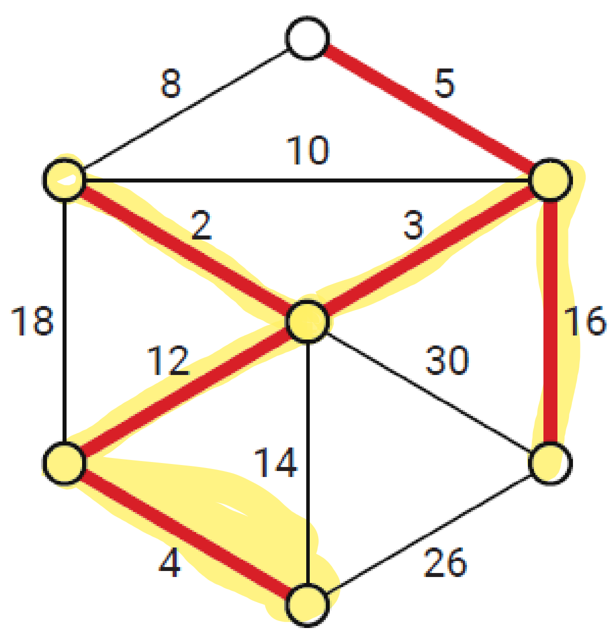


Figure 7.1. A weighted graph and its minimum spanning tree.

Cut property: Let $X \subseteq T$ where T is a MST in $G(V,E)$. Let $S \subset V$ such that no edge in X crosses between S and $V-S$; i.e. no edge in X has one endpoint in S and one endpoint in $V-S$. Among edges crossing between S and $V-S$, let e be an edge of minimum weight. Then $X \cup \{e\} \subseteq T'$ where T' is a MST in $G(V,E)$.

The cut property says that we can construct our tree *greedily*. Our greedy algorithms can simply take the minimum weight edge across two regions not yet connected. Eventually, if we keep acting in this greedy manner, we will arrive at the point where we have a minimum spanning tree. Although the idea of acting greedily at each point may seem quite intuitive, it is very unusual for such a strategy to actually lead to an optimal solution, as we will see when we examine other problems!

Proof: Suppose $e \notin T$. Adding e into T creates a unique cycle. We will remove a single edge e' from this unique cycle, thus getting $T' = T \cup \{e\} - \{e'\}$. It is easy to see that T' must be a tree — it is connected and has $n-1$ edges. Furthermore, as we shall show below, it is always possible to select an edge e' in the cycle such that it crosses between S and $V-S$. Now, since e is a minimum weight edge crossing between S and $V-S$, $w(e') \geq w(e)$. Therefore $w(T') = w(T) + w(e) - w(e') \leq w(T)$. However since T is a MST, it follows that T' is also a MST and $w(e) = w(e')$. Furthermore, since X has no edge crossing between S and $V-S$, it follows that $X \subseteq T'$ and thus $X \cup \{e\} \subseteq T'$.

How do we know that there is an edge $e' \neq e$ in the unique cycle created by adding e into T, such that e' crosses between S and $V-S$? This is easy to see, because as we trace the cycle, e crosses between S and $V-S$, and we must cross back along some other edge to return to the starting point. ■