CHAPTER 5

Trees

5.1. Forests, Trees and Leaves

A **forest** is an acyclic graph (that is, a graph with no cycles). The connected components of a forest are called **trees**. Therefore, a **tree** is a connected acyclic graph. In particular, any tree is a bipartite graph. A **leaf** is a vertex of degree one. In the figure below, we have a tree with seven leaves.

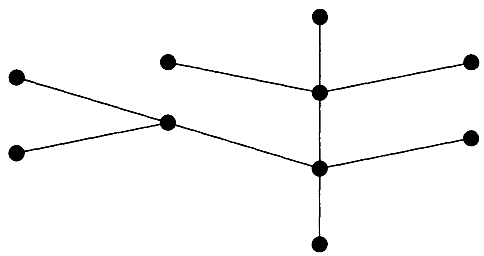

FIGURE 5.1

Given a graph X and a vertex v, we denote by $X - v$ the graph obtained by deleting the vertex v and any edges incident with v. We begin by proving the following:

LEMMA 5.1.1. *Every tree with $n \geq 2$ vertices has at least two leaves. Deleting a leaf from an n-vertex tree gives a tree with $n-1$ vertices.*

PROOF. A connected graph with at least two vertices has at least one edge. Let us consider a maximal path in the graph joining u and v (say). Every neighbour of u or v must be member of the path for otherwise, this would violate maximality of the path. If u or v had two neighbours, we would get cycle. Thus, u and v must be leaves. Now let v be a leaf. We will show that $X' = X - v$ is a tree. Clearly, $X - v$ is acyclic because deleting a vertex is not going to increase the number of cycles. We must show it is connected. Given two vertices in X', let P be a path joining them in X which exists because X is connected.

S. M. Cioabă and M. R. Murty, *A First Course in Graph Theory and Combinatorics*

This path cannot involve v for otherwise v will have degree at least two. Therefore, X' is connected. ∎

We now give the following characterization of trees.

THEOREM 5.1.2. *Let X be a graph on n vertices. The following are equivalent.*

(1) *X is a tree.*
(2) *X is connected and has $n-1$ edges.*
(3) *X has $n-1$ edges and no cycles.*
(4) *For any $u, v \in V(X)$, there is a unique path joining them.*

PROOF. To prove (1) implies (2), we use induction. By the previous lemma, let v be a leaf and consider the tree $X' = X - v$ with $n-1$ vertices. By induction, it has $n-2$ edges and together with the edge joining X' to v, we get a total of $n-1$ edges. The same argument shows that (1) implies (3). To prove that (2) implies (3), let us suppose X has a cycle. We may delete edges from any cycle until we get a graph X' which is acyclic and has n vertices. But then, X' is a tree and so has $n-1$ edges. Thus, no edges were deleted from X and X has no cycles. We can also show that (3) implies (1) as follows. Let $X_1, ..., X_k$ be the connected components of X. Since every vertex appears in one component, we have that

$$\sum_{i=1}^{k} |V(X_i)| = n.$$

As X has no cycles, each component is a tree so that $|E(X_i)| = |V(X_i)| - 1$ for each i. Thus, the number of edges of X is $n-k$. But as X has $n-1$ edges, $k=1$ and so X has only one connected component. Therefore, X is a tree. Finally, we must show the equivalence of (1) and (4). Clearly, (1) implies (4) for otherwise X would have a cycle. Conversely, if any two points have a unique path joining them, there are no cycles in the graph and moreover, X is connected. This completes the proof. ∎

5.2. Counting Labeled Trees

Arthur Cayley (1821-1895) spent 14 years as a lawyer during which he published 250 mathematical papers. In total, he published over 900 papers and notes covering almost every aspect of mathematics.

A classical result of Cayley states that the number of labeled trees on n vertices is n^{n-2}. Despite its simplicity, it is remarkable that there is no simple proof of this formula. We apply an inductive argument to deduce it.

Let $G(n,m)$ be the number of connected graphs on n labeled vertices and m edges. Let $F(n,m)$ denote the number of such graphs that have no vertices of degree 1. Let A be the set of connected graphs on n labeled vertices having m edges. Let A_i be the subset of A of connected graphs with vertex v_i of degree 1. Thus,

$$F(n,m) = |A \setminus \cup_i A_i|.$$

Let us observe that $|A_i| = G(n-1, m-1)(n-1)$ and generally

$$|A_I| = G(n-|I|, m-|I|)(n-|I|)^{|I|}.$$

Then, by the inclusion-exclusion principle, we have

$$F(n,m) = \sum_{I \subseteq V} (-1)^{|I|} G(n-|I|, m-|I|)(n-|I|)^{|I|}.$$

By collecting subsets of the same cardinality in the sum on the right, we obtain the following result.

THEOREM 5.2.1.

$$F(n,m) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} G(n-i, m-i)(n-i)^i.$$

Theorem 5.1.2 tells us that any connected graph on n vertices and $n-1$ edges is necessarily a tree. Thus, $G(n, n-1)$ is the number of labeled trees on n vertices. Since every tree has a leaf, we have that $F(n, n-1) = 0$.

THEOREM 5.2.2. *If T_n denotes the number of labeled trees on n vertices, then*

$$\sum_{i=0}^{n} (-1)^i \binom{n}{i} T_{n-i}(n-i)^i = 0.$$

Now we are ready to prove Cayley's formula.

THEOREM 5.2.3 (Cayley, 1889). *For $n \geq 2$,*

$$T_n = n^{n-2}.$$

PROOF. We prove the theorem by induction on n. For $n = 2$, the formula is clear. By induction, $T_{n-i} = (n-i)^{n-i-2}$ for $i \geq 1$. Using Theorem 5.2.2, we obtain that

$$T_n + \sum_{i=1}^{n} (-1)^i \binom{n}{i} (n-i)^{n-2} = 0.$$

By Theorem 3.3.1, the latter sum is $-n^{n-2}$ from which we deduce the theorem. ∎

5.3. Spanning Subgraphs

A **spanning subgraph** of a graph X is a subgraph with vertex set $V(X)$. A **spanning tree** is a spanning subgraph which is a tree. Given a graph X, we let $\tau(X)$ denote the number of spanning trees of X.

In a graph X, the graph obtained by deleting an edge e is denoted $X-e$. In this case, let us note that the vertices of e still belong to $X-e$. It may happen that this process increases the number of components of the graph, in which case we call e a **cut edge** or a **bridge**. The **contraction** of X by an edge e with endpoints u and v is the graph obtained by replacing u and v by a single vertex whose incident edges are the edges other than e that were incident to u or v. The resulting graph, denoted X/e has one less edge than X.

THEOREM 5.3.1. *If $\tau(X)$ is the number of spanning trees in X and $e \in E(X)$ is not a loop, then*

$$\tau(X) = \tau(X-e) + \tau(X/e).$$

PROOF. The spanning trees of X that omit e are counted by $\tau(X-e)$. The spanning trees that contain e are in one-to-one correspondence with the spanning trees of X/e. To see this, note that when we contract e in a spanning tree that contains e, we obtain a spanning tree of X/e because the resulting subgraph of X/e is spanning, connected and has the right number of edges. Since the other edges maintain their identity under contraction, no two trees are mapped to the same spanning tree of X/e by this operation. Also, each spanning tree of X arises in this way and so the function is a bijection. ∎

Recall that the Laplacian of a graph X is the matrix

$$L = D - A,$$

where A is the adjacency matrix of X and D is the diagonal matrix whose (i,i)-th entry equals the degree of vertex i.

Gustav Robert Kirchhoff (1824-1887) is perhaps best known for the Kirchhoff's laws in electrical circuits. These were announced in 1845 and extended previous work of Georg Simon Ohm (1789-1854).

A celebrated theorem of Kirchhoff from 1847 gives the number $\tau(X)$ via a determinant formula. This results is also known as the Matrix-Tree Theorem.

THEOREM 5.3.2 (Matrix-Tree Theorem). *For any loopless graph X, the number of spanning trees $\tau(X)$ equals $(-1)^{i+j}$ times the determinant of the matrix obtained by deleting the i-th row and j-th column of the Laplacian matrix L.*

We will not prove this theorem since it involves detailed linear algebra. We pause to remark that Cayley's theorem can be deduced easily from this more general result as follows. The number of trees on a vertex set $v_1, ..., v_n$ is the number spanning trees of the complete graph K_n. The adjacency matrix of K_n is $J - I$ with notation of the previous chapter. Thus, the Laplacian of the complete graph is

$$(n-1)I - (J - I)$$

and any cofactor is the determinant of $(n-1)I_{n-1} - (J_{n-1} - I_{n-1})$ where we have written the subscript to indicate the size of our matrix. By Example 4.2.3 in Chapter 4, we see that this determinant is the characteristic polynomial of the graph K_{n-1} evaluated at $\lambda = n - 1$ which is

$$[\lambda - (n-2)](\lambda + 1)^{n-2} = n^{n-2}$$

and thus, we recover Cayley's formula.

Theorem 5.3.2 can be stated in more succinct terms. Recall that the **classical adjoint** of a matrix A, denoted $\mathrm{adj}(A)$, is the transpose of the matrix whose i,j-th entry is $(-1)^{i+j}$ times the determinant of the matrix obtained from A by deleting the i-th row and j-th column. If J denotes (as before) the matrix all of whose entries are equal to 1, then Theorem 5.3.2 is equivalent to the assertion that

$$\mathrm{adj}(L) = \tau(X)J.$$

For example, Cayley's formula can be restated as

$$\mathrm{adj}(nI - J) = n^{n-2}J.$$

It is not hard to see that

$$J^2 = nJ, \qquad JL = LJ = 0.$$

These equations imply that $(nI - J)(J + L) = nJ - J^2 + nL - JL = nL$. Thus,

$$\mathrm{adj}(J + L)\mathrm{adj}(nI - J) = \mathrm{adj}((nI - J)(J + L)) = \mathrm{adj}(nL).$$

Cayley's formula implies $\mathrm{adj}(nI - J) = n^{n-2}J$. Also, $\mathrm{adj}(nL) = n^{n-1}\mathrm{adj}(L$ because in the adjoint the entries are formed by taking $(n-1) \times (n-1)$ determinants. We therefore deduce that

$$[\mathrm{adj}(J + L)]J = n\mathrm{adj}(L).$$

By Theorem 5.3.2,

$$\mathrm{adj}(L) = \tau(X)J$$

so we obtain

$$[\mathrm{adj}(J + L)]J = n\tau(X)J.$$

Multiplying both sides of the equation by $(J+L)$ on the left gives

$$[\det(J+L)]J = n\tau(X)(J+L)J.$$

Because $(J+L)J = J^2 + LJ = nJ$, we therefore deduce the next result.

THEOREM 5.3.3. *Let X be a simple graph whose Laplacian matrix is L. The number of spanning trees in X is given by*

$$\tau(X) = n^{-2}\det(J+L).$$

In the case X is a connected k-regular graph, we can derive a nicer formula. Recall that the adjacency matrix of X has eigenvalue k. Since X is connected, the multiplicity of this eigenvalue is 1. To see this, let $v = (x_1, ..., x_n)$ be an eigenvector corresponding to the eigenvalue k. The equation

$$A_X v = kv$$

implies that

$$\sum_{j=1}^{n} a_{ij}x_j = kx_i.$$

Without any loss of generality suppose that $x_1 > 0$ and that $x_1 = \max_{1\le i\le n} x_i$. If for some i, $x_i < x_1$, then

$$kx_1 = \sum_{j=1}^{n} a_{1j}x_j < kx_1$$

which is a contradiction. Thus, all the x_i are equal and so every eigenvector must be a multiple of $(1,1,...,1)$. If X is not connected, the multiplicity of the eigenvalue is easily seen to be the number of connected components. By Theorem 5.3.3, we must compute the determinant

$$\det(J + kI - A)$$

which is just the characteristic polynomial of $A-J$ evaluated at k. The eigenvalues of $A-J$ are easily determined. Let $v_1, ..., v_n$ be a orthogonal basis of eigenvectors of A, with v_1 a multiple of $(1,1,...,1)$ corresponding to the eigenvalue k. Then, for $2 \le i \le n$,

$$(A-J)v_i = \lambda_i v_i$$

as $Jv_i = 0$. This is true because v_1 is orthogonal to the v_i. Also,

$$(A-J)v_1 = (k-n)v_1$$

so this determines all the eigenvalues of $A-J$ and their multiplicity. The characteristic polynomial of $A-J$ is

$$(\lambda-(k-n))\prod_{i=2}^{n}(\lambda-\lambda_i).$$

Putting this together with Theorem 5.3.3 gives

THEOREM 5.3.4. *If X is a connected k-regular graph, then the number of spanning trees of X is given by*

$$\frac{\prod_{i=2}^{n}(k-\lambda_i)}{n},$$

where the product is over the eigenvalues unequal to k.

This theorem can, for instance, be used to compute the number of spanning trees of the bipartite graph $K_{n,n}$ (see Exercise 5.5.11).

5.4. Minimum Spanning Trees and Kruskal's Algorithm

In many contexts in which graph theory is applied, we consider **weighted graphs**. That is, we suppose we have a graph X together with a "weight" function $w : E(X) \to \mathbb{R}_+$ that assigns to each edge a positive weight. For example, our graph could be a network of cities, and the weight function could be the cost of putting a communication network between the two cities. We will be interested in finding a connected subgraph so that its total "cost", i.e., the sum of the weights of the edges in the subgraph, is minimal. Clearly, if there is a cycle, we can delete a 'costly' edge from the cycle and so, what we are searching is a spanning tree whose 'cost' is minimal. We call such a tree a **minimum spanning tree**. Of course, it need not be unique.

There is a fundamental algorithm, called **Kruskal's algorithm** which determines a minimum spanning tree of any connected graph in a 'greedy' fashion. It can be described as follows. Choose an edge e_1 of X with $w(e_1)$ minimal. Eliminate it from the list. Inductively choose $e_2, ..., e_{n-1}$ in the same manner subject to the constraint that the newly chosen edge does not form a cycle with previously chosen edges. The required spanning tree is the subgraph with these edges. Before we prove that this **greedy algorithm** actually works, we illustrate this with an example.

Consider the following weighted adjacency matrix giving the cost of building a road from one city to another. An infinite entry indicates there is a mountain in the way and a road cannot be built. The question is to determine the least cost of making all the cities reachable from each other. This amounts to finding a spanning tree with minimum "length".

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array}\begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 3 & 5 & 11 & 9 \\ 3 & 0 & 3 & 9 & 8 \\ 5 & 3 & 0 & \infty & 10 \\ 11 & 9 & \infty & 0 & 7 \\ 9 & 8 & 10 & 7 & 0 \end{pmatrix} \end{array}.$$

The algorithm proceeds first by finding an edge of minimum weight, AB say. It then deletes this edge. In the next step, the algorithm finds the next smallest entry, BC say. The algorithm continues in this way and whenever an edge is chosen which produces a cycle, the algorithm does not select it. Thus, in the example below, AC is the next smallest entry but we would not choose it for it produces a cycle with AB and BC.

Thus the next entry to choose is DE followed by BE. Thus the minimum spanning tree is given in Figure 5.2. The minimum 'cost' is 21.

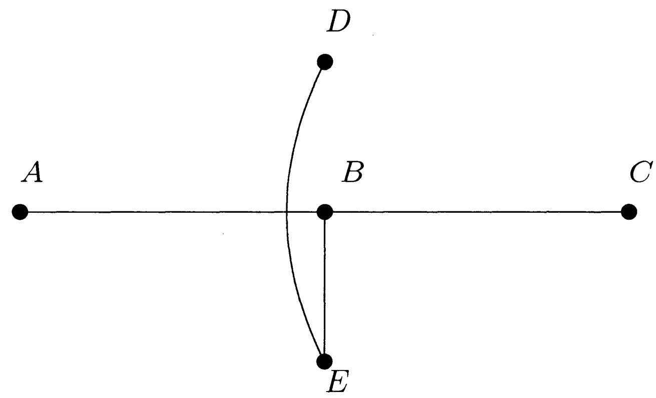

FIGURE 5.2. A minimum spanning tree of weight 21

THEOREM 5.4.1. *In a weighted connected graph X, Kruskal's algorithm constructs a minimum weight spanning tree.*

PROOF. Kruskal's algorithm produces a tree since it selects $n - 1$ edges which do not form cycles from a connected graph on n vertices. Let T be the tree produced by the algorithm and let T^* be a minimum weight spanning tree. If $T = T^*$, we are done. If not, let e be the first edge chosen for T that is not in T^*. Adding e to T^* creates a cycle C

since T^* is a spanning tree. Because T contains no cycles, we deduce that the cycle C must contain at least one edge e' not in $E(T)$. Now consider the subgraph $T^* + e - e'$ of X obtained from T^* by adding the edge e and removing the edge e'. The subgraph $T^* + e - e'$ is actually a spanning tree of X because it has $n - 1$ edges and contains no cycles. Since T^* contains e' and all the edges of T chosen before e, it means that both e' and e are available when the algorithm chooses e and therefore, $w(e) \leq w(e')$. Thus, $T^* + e - e'$ is a spanning tree with weight at most that of T^* (actually with the same weight as T^* since T^* is a minimum weight spanning tree) that agrees with T for a longer initial list of edges than T^* does. Repeating this process, we deduce that the tree created by Kruksal's algorithm has the same weight as T^* which finishes the proof. ■

5.5. Exercises

EXERCISE 5.5.1. Prove that in any tree, every edge is a bridge.

EXERCISE 5.5.2. Let X be a connected graph on n vertices. Show that X has exactly one cycle if and only if X has n edges. Prove that a graph with n vertices and e edges contains at least $e - n + 1$ cycles.

EXERCISE 5.5.3. Let $d_1, d_2, ..., d_n$ be positive integers. Show that there exists a tree on n vertices with vertex degrees $d_1, d_2, ..., d_n$ if and only if

$$\sum_{i=1}^{n} d_i = 2n - 2.$$

EXERCISE 5.5.4. The number of trees with degree sequence $d_1, \ldots, d_n$ with $d_1 + \cdots + d_n = 2n - 2$ is

$$\binom{n-2}{d_1 - 1, \ldots, d_n - 1} = \frac{(n-2)!}{(d_1 - 1)! \ldots (d_n - 1)!}.$$

EXERCISE 5.5.5. Show that if X is a tree on n labeled vertices, then each element of $\{X - e : e \in E(X)\}$ is a forest of two trees.

EXERCISE 5.5.6. Let T and T' be two distinct trees on the same set of n vertices. Show that for each edge $e \in E(T) \setminus E(T')$, there exists $e' \in E(T') \setminus E(T)$ such that $T \setminus \{e\} \cup \{e'\}$ is a tree.

EXERCISE 5.5.7. Let T_n be the number of trees on n labeled vertices. Prove that

$$2(n-1)T_n = \sum_{i=1}^{n-1} \binom{n}{i} T_i T_{n-i} i(n-i).$$

EXERCISE 5.5.8. Show that

$$\sum_{i=1}^{n-1} \binom{n}{i} i^{i-1}(n-i)^{n-i-1} = 2(n-1)n^{n-2}.$$

EXERCISE 5.5.9. Let $G(r, s; m)$ be the number of connected bipartite graphs with partite sets of size r and s having m edges, and let $F(r, s; m)$ be the number of such graphs not containing any vertices of degree 1. Prove that

$$F(r, s; m) = \sum_{i,j} \binom{r}{i}\binom{s}{j}(-1)^{i+j} G(r-i, s-j; m-i-j)(s-j)^i(r-i)^j.$$

EXERCISE 5.5.10. Putting $m = r + s - 1$ in the previous exercise, notice that $G(r, s; r+s-1)$ counts the number $T(r, s)$ (say) of spanning trees in the bipartite graph $K_{r,s}$. Deduce that

$$0 = \sum_{i,j} \binom{r}{i}\binom{s}{j}(-1)^{i+j} T(r-i, s-j)(s-j)^i(r-i)^j$$

and that $T(r, s) = r^{s-1}s^{r-1}$.

EXERCISE 5.5.11. Show that the number of spanning trees of the bipartite graph $K_{n,n}$ is n^{2n-2}.

EXERCISE 5.5.12. The **Wiener index** of a graph X is $W(X) = \sum_{u,v \in V(X)} d(u, v)$, where $d(u, v)$ denotes the distance from u to v. Show that if X is a tree on n vertices, then

$$W(K_{1,n-1}) \leq W(X) \leq W(P_n).$$

EXERCISE 5.5.13. A communication link is desired between five universities in Canada: Queen's, Toronto, Waterloo, McGill and UBC. With obvious notation, the matrix below gives the cost (in thousands of dollars) of building such a connection between any two of the universities.

$$\begin{array}{c} \\ Q \\ T \\ W \\ M \\ U \end{array} \begin{array}{c} \begin{array}{ccccc} Q & T & W & M & U \end{array} \\ \begin{pmatrix} - & 350 & 400 & 300 & 1200 \\ 350 & - & 100 & 600 & 1300 \\ 400 & 100 & - & 700 & 1400 \\ 300 & 600 & 700 & - & 1600 \\ 1200 & 1300 & 1400 & 1600 & - \end{pmatrix} \end{array}.$$

Use the greedy algorithm to determine the minimal cost so that all universities are connected.

EXERCISE 5.5.14. Every tree with maximum degree d has at least d leaves. Construct a tree with n vertices and maximum degree d for each $n > d \geq 2$.

EXERCISE 5.5.15. Let X be a graph on $n \geq 3$ vertices such that by deleting any vertex of X, we obtain a tree. Find X.

EXERCISE 5.5.16. Show that every connected graph X contains at least two vertices u with the property that $X \setminus \{u\}$ is connected. What are the trees on n vertices that contain exactly two vertices with this property ?

EXERCISE 5.5.17. Show that the graph obtained from K_n by removing one edge has $(n-2)n^{n-3}$ spanning trees.

EXERCISE 5.5.18. Let G_n be the graph obtained from the path P_n by adding one vertex adjacent to all the vertices of the path P_n. Determine the number of spanning trees of G_n.

EXERCISE 5.5.19. If G is a graph on n vertices having maximum degree $k \geq 2$ and diameter D, show that

$$n \leq \begin{cases} 2D+1, \text{ if } k=2 \\ \frac{k[(k-1)^D-1]}{k-2}+1, \text{ otherwise.} \end{cases}$$

EXERCISE 5.5.20. The **center** of a graph X is the subgraph induced by the vertices of minimum eccentricity. Show that the center of a tree is a vertex or an edge.