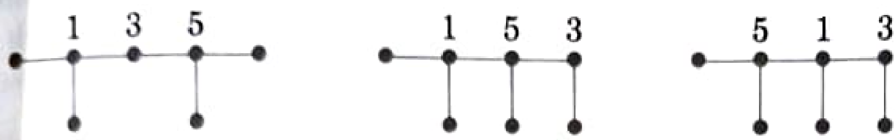
2.2.5. Example. *Trees with fixed degrees.* Consider trees with vertices $\{1,2,3,4,5,6,7\}$ that have degrees (3,1,2,1,3,1,1), respectively. We compute $\frac{(n-2)!}{\prod(d_i-1)!}=30$; the trees are suggested below. There are six ways to complete the first tree (pick from the remaining four vertices the two adjacent to vertex 1) and twelve ways to complete each of the others (pick the neighbor of vertex 3 from the remaining four, and then pick the neighbor of the central vertex from the remaining three). □



When $\Sigma n_i = n$, the quantity $\frac{n!}{\prod n_i!}$ is called the *multinomial coefficient* $\binom{n}{n_1,\dots,n_k}$, because it is the coefficient of $\prod(x_i^{n_i})$ in the expansion of $(\Sigma_{i=1}^k x_i)^n$. The contributions to the coefficient of this term correspond to n-tuples that are arrangements of the n letters consisting of n_i letters of type i. When we set $x_i=1$ for all i, this tells us that the total number of n-tuples formed from k types of letters, over all possible multiplicities, is k^n, which agrees with Cayley's formula.

SPANNING TREES IN GRAPHS

Cayley's formula also follows from the more general Matrix Tree Theorem implicit in earlier work of Kirchhoff [1847]. This provides a formula that counts the spanning subtrees of any loopless graph G; Cayley's formula results when $G = K_n$ (Exercise 15). We first describe a recursive way to count spanning trees, by separately counting those that contain a particular edge e and those that omit e.

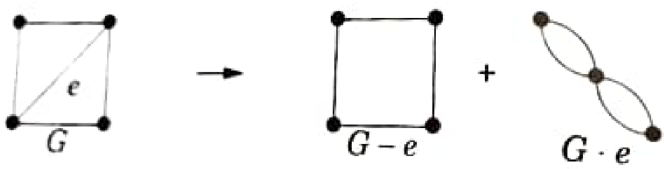
2.2.6. Definition. If $e = uv$ is an edge of G, then *contraction* of e is the operation of replacing u and v by a single vertex whose incident edges are the edges other than e that were incident to u or v. The resulting graph, denoted $G \cdot e$, has one less edge than G.

Visually, we think of contracting e as shrinking e to a single point. Edge contraction edge can produce multiple edges. To count spanning trees correctly, we must keep the multiple edges (the example below shows why), but in other applications of contraction the multiple edges may be irrelevant. On the other hand, we may freely discard loops that arise in contraction, because no spanning tree contains a loop. The recurrence applies for all graphs.

2.2.7. Proposition. If $\tau(G)$ denotes the number of spanning trees of a graph G and $e \in E(G)$, then $\tau(G) = \tau(G - e) + \tau(G \cdot e)$.

Proof. The spanning trees of G that omit e are precisely the spanning trees of $G - e$. The number of spanning trees that contain e is $\tau(G \cdot e)$, because there is a natural bijection between spanning trees of $G \cdot e$ and spanning trees of G that contain e. Contracting e in a spanning tree of G that contains e yields a spanning tree of $G \cdot e$. The other edges maintain their identity under contraction, so no two trees collapse onto the same spanning tree of $G \cdot e$ via this operation. Furthermore, each spanning tree of $G \cdot e$ arises in this way. Hence the map is a bijection. □

2.2.8. Example. *A step in the recurrence.* The graphs on the right each have four spanning trees, so the recurrence for spanning trees implies that the graph on the left has eight spanning trees. □



Computation using the recurrence requires initial conditions for graphs with no edges. If one vertex remains, there is one spanning tree. If more than one vertex remains, there is no spanning tree. If the computer follows the recurrence by deleting or contracting every edge, it computes $2^{e(G)}$ terms. This can be reduced by ignoring loops and by recognizing special graphs G where we know $\tau(G)$.

2.2.9. Remark. If G is a connected loopless graph with no cycle other than repeated edges, then $\tau(G)$ is the product of the edge multiplicities. A disconnected graph has no spanning trees. □

Despite such reductions, the recursive computation is impractical for large graphs. The Matrix Tree Theorem computes $\tau(G)$ using a determinant. Determinants of n by n matrices can be computed using fewer than n^3 operations (for large n), which is much faster than $2^{e(G)}$. Again we delete loops before the computation since they don't affect spanning trees. The proof of the Matrix Tree Theorem requires matrix multiplication (and determinants).

2.2.10. Example. *A Matrix Tree computation.* The Matrix Tree Theorem below instructs us to form a matrix with the vertex degrees on the diagonal, subtract the adjacency matrix, delete a row and a column, and take the determinant. For the graph $K_4 - e$ in the example above, the vertex degrees are 3,3,2,2, so we form the matrix on the left below and take the determinant of the matrix in the middle. The result is the number of spanning trees! □

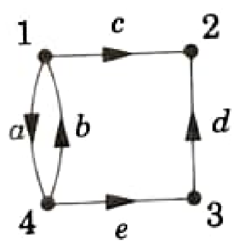
$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix} \rightarrow 8$$

2.2.11. Theorem. (Matrix Tree Theorem). Given a loopless graph G with vertex set $v_1, \ldots, v_n$, let a_{ij} be the number of edges of the form $v_i v_j$. Let Q be the matrix with entry (i, j) being $-a_{ij}$ when $i \neq j$ and $d(v_i)$ when $i = j$. If Q^* is the matrix obtained by deleting any row s and column t of Q, then $\tau(G) = (-1)^{s+t} \det Q^*$.

Proof (optional). We prove this only when $s = t$; the general statement then follows using Exercise 18.

Step 1. If G' is an orientation of G, and M is the incidence matrix of G', then $Q = MM^T$. If the directed edges are $e_1, \ldots, e_m$, then the entries of M are $m_{ij} = 1$ if v_i is the tail of e_j, $m_{ij} = -1$ if v_i is the head of e_j, and $m_{ij} = 0$ if v_i does not belong to e_j. Since every entry in the n by n matrix MM^T is the dot product of rows of M, off-diagonal entries in the product count -1 for every edge of G between the two vertices, and diagonal entries count vertex degrees.

$$M = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline 1 & -1 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & -1 & -1 & 0 \\ 3 & 0 & 0 & 0 & 1 & -1 \\ 4 & 1 & -1 & 0 & 0 & 1 \end{array}$$



$$Q = \begin{pmatrix} 3 & -1 & 0 & -2 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -2 & 0 & -1 & 3 \end{pmatrix}$$

Step 2. If B is an $(n-1)\times(n-1)$ submatrix of M, then $\det B = 0$ if the corresponding $n-1$ edges contain a cycle, and $\det B = \pm 1$ if they form a spanning tree of G. If the edges corresponding to the columns contain a cycle C, then the columns sum to the zero vector when weighted with $+1$ or -1 according as the directed edge is followed forward or backward when following the cycle. This equation of dependence yields $\det B = 0$.

For the other case, we use induction on n. For $n = 1$, by convention a 0×0 matrix has determinant 1. Suppose $n > 1$, and let T be the spanning tree whose edges are the columns of B. Since T has at least two leaves, B contains a row corresponding to a leaf x of T. This row has only one nonzero entry in B. When computing the determinant by expanding along that row, the only submatrix B' given nonzero weight in the expansion corresponds to the spanning subtree of $G - x$ obtained by deleting x and its incident edge from T. Since B' is an $(n-2) \times$

$(n-2)$ submatrix of the incidence matrix for an orientation of $G - x$, the induction hypothesis implies that the determinant of B' is ± 1, and multiplying it by ± 1 gives the same result for B.

Step 3. Computation of $\det Q^*$. Let M^* be the matrix obtaining by deleting row i of M, so $Q^* = M^*(M^*)^T$. We may assume $m \geq n-1$, else both sides have determinant 0 and there are no spanning subtrees. The Binet-Cauchy formula expresses the determinant of a product of matrices, not necessarily square, in terms of the determinants of submatrices of the factors. In particular, if $m \geq p$, A is a $p \times m$ matrix, and B is an $m \times p$ matrix, then $\det AB = \Sigma_S \det A_S \det B_S$, where the summation runs over all $S \subseteq [m]$ consisting of p indices, A_S is the submatrix of A having the columns indexed by S, and B_S is the submatrix of B having the rows indexed by S (Exercise 19). When we apply the Binet-Cauchy formula to $Q^* = M^*(M^*)^T$, the submatrix A_S is an $(n-1)\times(n-1)$ submatrix of M as discussed in Step 2, and $B_S = A_S^T$. Hence the summation counts $1 = (\pm 1)^2$ for each set of $n-1$ edges corresponding to a spanning tree and 0 for each other set of $n-1$ edges. □

Tutte extended this theorem to directed graphs. His theorem reduces to the Matrix Tree Theorem when the digraph is symmetric; a digraph is *symmetric* if its adjacency matrix is symmetric.

2.2.12. Definition. A *branching* or *out-tree* is an orientation of a tree having a root of in-degree 0 and all other vertices of indegree 1. An *in-tree* is an out-tree with its edges reversed. Given a digraph G, let $Q^- = D^- - A'$ and $Q^+ = D^+ - A'$, where D^-, D^+ are the diagonal matrices of in-degrees and out-degrees in G, and the i,j-entry of A' is the number of edges from v_j to v_i.

2.2.13. Theorem. (Directed Matrix Tree Theorem - Tutte [1948]) In a digraph, with Q^- and Q^+ defined as above, the number of out-trees (in-trees) rooted at v_i is the value of any cofactor in the ith row of Q^- (ith column of Q^+). □

2.2.14. Example. The digraph below has two out-trees rooted at 1 and two in-trees rooted at 3. The determinants behave as claimed. □

$$Q^+ = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix}$$

$$Q^- = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 2 \end{pmatrix}$$

DECOMPOSITION AND GRACEFUL LABELINGS

A *decomposition* of a graph G is a partition of $E(G)$ into pairwise edge-disjoint subgraphs. We can always decompose G into single edges, so we might ask whether we can decompose G into isomorphic copies of a larger tree T. This requires that $e(G)$ be a multiple of $e(T)$; is that also sufficient? If G is regular, the answer is "maybe". Häggkvist conjectured that if G is a $2m$-regular graph and T is a tree with m edges, then $E(G)$ can be partitioned into $n(G)$ copies of T. Even the "simplest" case when G is a clique is still open and notorious.

2.2.15. Conjecture. (Ringel [1964]) If T is a fixed tree with m edges, then K_{2m+1} can be decomposed into $2m+1$ copies of T. □

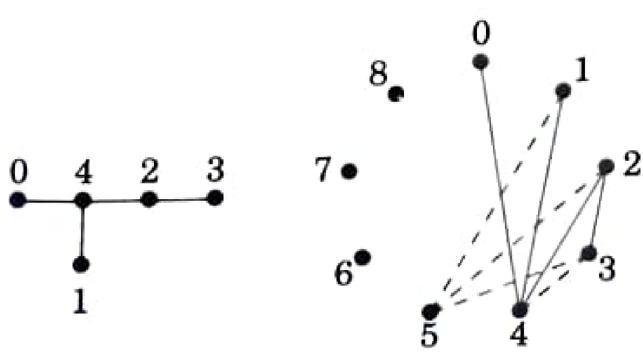
Attempts to prove Ringel's conjecture have focused on a stronger conjecture about trees, called the *Graceful Tree Conjecture*. This conjecture implies Ringel's conjecture and a similar statement about decomposing cliques of even order (Exercise 25).

2.2.16. Conjecture. (Graceful Tree Conjecture - Kotzig, Ringel [1964]) If T is a tree with m edges, then the vertices of T can be given the distinct numbers $0,\dots,m$ in such a way that the edge-differences are $\{1,\dots,m\}$. Such a numbering is called a *graceful labeling*. □

2.2.17. Theorem. (Rosa [1967]) If a tree T with m edges has a graceful labeling, then K_{2m+1} has a decomposition into $2m+1$ copies of T.

Proof. View the vertices of K_{2m+1} as the congruence classes modulo $2m+1$. The *displacement* between two congruence classes is the number of unit moves needed to get from one to the other; the maximum displacement between two congruence classes modulo $2m+1$ is m. The edges of K_{2m+1} consist of m "displacement classes", each of size $2m+1$.

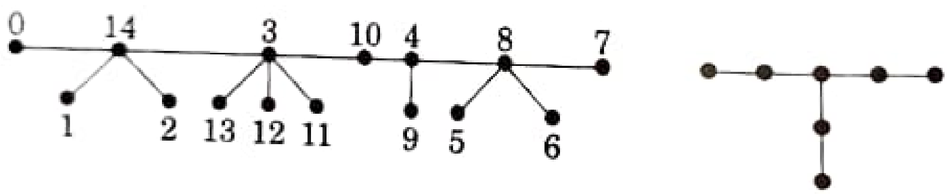
From a graceful labeling of T, we define copies of T in K_{2m+1} for $0 \le k \le 2m$. In the kth copy, the vertices are $k,\dots,k+m$ mod $2m+1$, with $k+i$ adjacent to $k+j$ if and only if i is adjacent to j in the graceful labeling. The 0th copy of T looks just like the graceful labeling and has



one edge with each displacement. Moving to the next copy shifts each edge to the next edge in its displacement class. Hence the $2m+1$ copies of T cycle through the $2m+1$ edges from each displacement class, without repetitions, and these $2m+1$ copies of T decompose K_{2m+1}. □

Graceful labelings are known to exist for some types of trees. In some ways, the *stars* ($K_{1,n-1}$) and the paths (P_n) are the simplest trees; stars minimize the diameter and paths minimize the maximum degree. We can obtain more general trees than stars by considering diameter at most k, for some fixed k. To generalize paths, we permit the addition of edges incident to a path, obtaining a class that includes the stars and the paths and has graceful labelings.

2.2.18. Example. *Graceful labeling of caterpillars.* A *caterpillar* is a tree having a path that contains at least one vertex of every edge (it can be taken to be a path of maximum length). The illustration shows a caterpillar with a graceful labeling and a tree that is not a caterpillar. Every caterpillar has a graceful labeling (Exercise 29). A *lobster* is a tree having a path from which every vertex has distance at most 2 (caterpillars with longer legs); it is not yet known whether all lobsters are graceful. □



2.2.19. Theorem. The following conditions on a tree G are equivalent and characterize the class of *caterpillars*.
A) G has a path incident to every edge.
B) Every vertex of G has at most two non-leaf neighbors.
C) G does not contain the tree on the right above.

Proof. Let G' denote the tree obtained from G by deleting each leaf of G. Condition A states that G' is a path, which is equivalent to $\Delta(G') \le 2$. Since the non-leaf neighbors of each non-leaf vertex remain in G', $\Delta(G') \le 2$ is also equivalent to condition B, and we have proved A⇔B. For B⇔C, G has a vertex with three non-leaf neighbors if and only if G has the forbidden subtree. □

EXERCISES

2.2.1. (−) Prove that the n-vertex graph $K_1 \vee C_{n-1}$ has a spanning tree with diameter k for each $k \in \{2, \dots, n-1\}$.