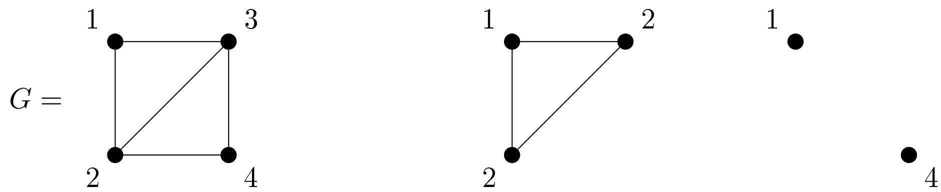
Contents

Example 1.44.



clique in G independent set in G

Definition 1.45. Let $G = (V, E)$ be a graph. The *clique number* $\omega(G)$ denotes the maximum number of vertices in a clique in G. The *independence number* let $\alpha(G)$ denote the number of vertices in a maximum-size independent set in G.

Example 1.46. In Example 1.44, $\omega(G) = 3$ and $\alpha(G) = 2$.

Claim 1.47. A vertex set $U \subseteq V(G)$ is a clique if and only if $U \subseteq V\left(\overline{G}\right)$ is an independent set.

Corollary 1.48. *We have* $\omega(G) = \alpha\left(\overline{G}\right)$ *and* $\alpha(G) = \omega\left(\overline{G}\right)$.

2 Trees

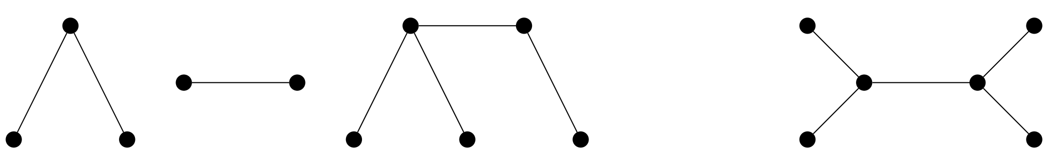
As previously mentioned, in several applications, one of the key properties of a graph is that it should be connected — that is, one should be able to travel from any one vertex to any other. In this chapter, we will take a closer look at connected graphs with as few edges as possible.

2.1 Trees

We begin with some central definitions, which are depicted in the examples below.

Definition 2.1. A graph having no cycle is *acyclic*, and an acyclic graph is called a *forest*. A *tree* is a connected acyclic graph, and a *leaf* (or *pendant vertex*) is a vertex of degree 1.

Example 2.2.

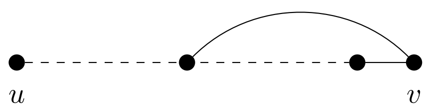


A forest A tree

Our first result shows that every tree must contain leaves.

Lemma 2.3. *Every finite tree with at least two vertices has at least two leaves. Furthermore, deleting a leaf from an n-vertex tree produces a tree with $n-1$ vertices.*

Proof. Let P be a maximal path (by inclusion) in the tree. Since the tree is connected and has at least two vertices, P must have positive length. Let $v \in V(P)$ be an endpoint of the path. By the maximality of P, the neighbours of v must all lie on the path P, as otherwise we could extend P to a longer path. If v had two or more neighbours on P, then we would obtain a cycle, contradicting the tree being acyclic. Hence, v must have degree one. Thus, the two endpoints of P provide the two desired leaves.



(If v had multiple neighbours on the path we would get a cycle).

Suppose v is a leaf of a tree G, and let $G' = G \backslash v$, the subgraph obtained by deleting v and its incident edge. If $u, w \in V(G')$, then no u, w-path P in G can pass through the vertex v of degree 1, so P is also present in G'. Hence, G' is connected. Since deleting a vertex cannot create a cycle, G' is also acyclic. We conclude that G' is a tree with $n-1$ vertices. $\square$

The last part of Lemma 2.3 is particularly important, as it allows us to prove statements about trees by induction on the number of vertices.

2.2 Equivalent definitions of trees

We defined trees as connected acyclic graphs. The following theorem provides some alternative descriptions of trees.

Theorem 2.4. *For an n-vertex simple graph G (with $n \geq 1$), the following are equivalent (and characterise the trees with n vertices).*

(a) G is connected and has no cycles.

(b) G is connected and has $n-1$ edges.

(c) G has $n-1$ edges and no cycles.

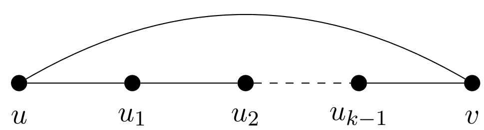
(d) For every pair $u, v \in V(G)$, there is exactly one u, v-path in G.

To prove this theorem, we will need a small lemma.

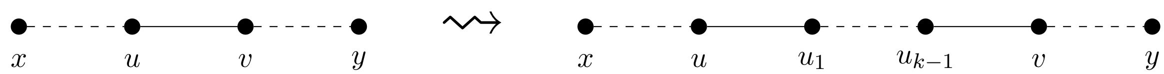
Definition 2.5. An edge of a graph is a *cut-edge* if its deletion disconnects the graph.

Lemma 2.6. *An edge contained in a cycle is not a cut-edge.*

Proof. Let $\{u, v\}$ belong to a cycle.



Then any path $x \dots y$ in G which uses the edge $\{u, v\}$ can be extended to a walk in $G \setminus \{u, v\}$ by using the rest of the cycle, as shown below:



By Proposition 1.33, there is an x, y-path within this walk, and so we deduce that $G \setminus \{u, v\}$ is still connected. □

With this lemma, we can now prove the theorem, showing that the different characterisations of trees are all equivalent.

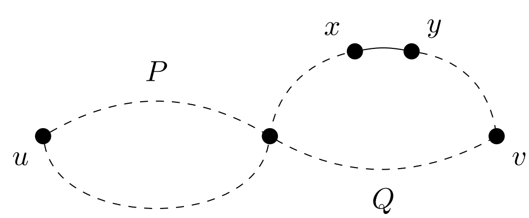
Proof of Theorem 2.4. We first demonstrate the equivalence of (a), (b) and (c) by proving that any two of {connected, acyclic, $n-1$ edges} implies the third.

(a) $\implies$ (b), (c): We use induction on n. For $n = 1$, an acyclic 1-vertex graph has no edge. For the induction step, suppose $n > 1$, and suppose the implication holds for graphs with fewer than n vertices. Given G, Lemma 2.3 provides a leaf v and states that $G' = G \backslash v$ is acyclic and connected. Applying the induction hypothesis to G' yields $e(G') = n - 2$, and hence, since v is incident to a single edge, $e(G) = n - 1$.

(b) $\implies$ (a), (c): Delete edges from cycles of G one by one until the resulting graph G' is acyclic. By Lemma 2.6, G' is still connected. By the paragraph above, G' has $n-1$ edges. Since this equals $|E(G)|$, no edges were deleted, and G itself is acyclic.

(c) $\implies$ (a), (b): Suppose G has k components with orders $n_1, \dots, n_k$. Since G has no cycles, each component satisfies property (a), and by the first paragraph the ith component has $n_i - 1$ edges. Summing this over all components yields $e(G) = \sum(n_i - 1) = n - k$. We are given $e(G) = n - 1$, so $k = 1$, and G is connected.

(a) $\implies$ (d): Since G is connected, G has at least one u, v-path for each pair $u, v \in V(G)$. Suppose G has distinct u, v-paths P and Q. Let $e = \{x, y\}$ be an edge in P but not in Q. The concatenation of P with the reverse of Q is a closed walk in which e appears exactly once. Hence, $(P \cup Q) \backslash e$ is an x, y-walk not containing e. By Proposition 1.33, this contains an x, y-path, which completes a cycle with e and contradicts the hypothesis that G is acyclic. Hence G has exactly one u, v-path.



(d) $\implies$ (a): If there is a u,v-path for every $u, v \in V(G)$, then G is connected. If G has a cycle C, then G has two paths between any pair of vertices on C. □

Definition 2.7. Given a connected graph G, a *spanning tree* T is a subgraph of G which is a tree and contains every vertex of G.

This establishes that the different characterisations of trees are indeed equivalent, and hence, for any application, we can choose the most suitable one. We conclude by collecting a few more facts about trees.

Corollary 2.8.

(a) Every connected graph on n vertices has at least $n-1$ edges and contains a spanning tree;

(b) Every edge of a tree is a cut-edge;

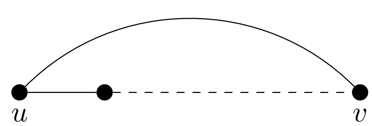
(c) Adding an edge to a tree creates exactly one cycle.

Proof.

(a) Delete edges from cycles of G one by one until the resulting graph G' is acyclic. By Lemma 2.6, G' is connected. The resulting graph is acylic so it is a tree, and contains $n-1$ edges. Therefore G had at least $n-1$ edges and contained a spanning tree.

(b) Note that deleting an edge from a tree T on n vertices leaves $n-2$ edges, so the graph is disconnected by (a).

(c) Let $u, v \in T$. There is a unique path in T between u and v, so adding an edge (u, v) closes this path to a unique cycle.



□

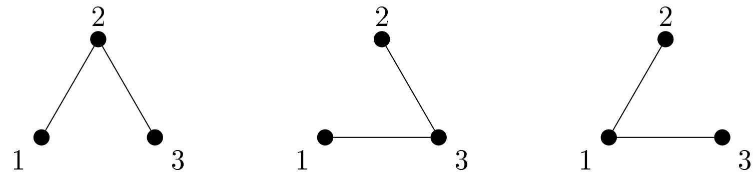
2.3 Cayley's formula – Prüfer code

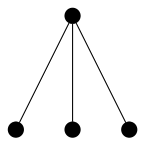
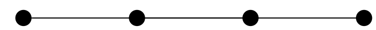
Now that we know what a tree is, we turn our attention to determining how many there are.

Question 2.9. What is the number of spanning trees in a labelled complete graph on n vertices?

Example 2.10.

$n = 3$:



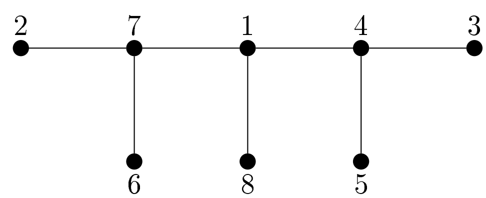
$n = 4$: 4 stars  $12 = 4!/2$ paths 

Theorem 2.11 (Cayley's Formula)**.** *There are n^{n-2} trees with vertex set $[n]$.*

We shall give two proofs of Cayley's formula. In our first proof, we establish a bijection between trees on $[n]$ and sequences in $[n]^{n-2}$.

Definition 2.12 (Prüfer code)**.** Let T be a tree on an ordered set S of n vertices. To compute the Prüfer sequence $f(T)$, iteratively delete the leaf with the smallest label and append the label of its *neighbour* to the sequence. After $n-2$ iterations a single edge remains and we have produced a sequence $f(T)$ of length $n-2$.

Example 2.13.

$T =$ 

We compute the Prüfer code for T as follows:

- Delete leaf 2, append 7 to the code
- Delete leaf 3, append 4 to the code
- Delete leaf 5, append 4 to the code
- Deleat leaf 4, append 1 to the code
- Delete leaf 6, append 7 to the code
- Deleat leaf 7, append 1 to the code

The edge remaining is $(1, 8)$. We then have $f(T) = (7, 4, 4, 1, 7, 1)$.

Proposition 2.14. *For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.*

Proof. We need to show every sequence $(a_1, \dots, a_{n-2}) \in S^{n-2}$ defines a unique tree T such that $f(T) = (a_1, \dots, a_{n-2})$. If $n = 2$, then there is exactly one tree on 2 vertices and the algorithm defining f always outputs the empty sequence, the only sequence of length zero. So the claim clearly holds for $n = 2$.

Now, assume $n > 2$ and the claim holds for all ordered vertex sets S' of size less than n. Consider a sequence $(a_1, \dots, a_{n-2}) \in S^{n-2}$. We need to show that $(a_1, \dots, a_{n-2})$ can be uniquely produced by the algorithm.

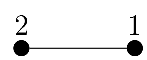
Suppose that the algorithm produces $f(T) = (a_1, \dots, a_{n-2})$ for some tree T. Then the vertices $\{a_1, \dots, a_{n-2}\}$ are precisely those that are not a leaf in T. Indeed, if a vertex v is a leaf in T then it can only appear in $f(T)$ if its neighbour gets deleted during the algorithm. But this would leave v as an isolated vertex, which is impossible. Conversely, if a vertex v is not a leaf then one of its neighbours must be deleted during the algorithm (it cannot be itself deleted before this happens). When this neighbour of v is deleted, v will be added to the Prüfer code for T, so is in $\{a_1, \dots, a_{n-2}\}$.

This implies that the label of the first leaf removed from T is the minimum element of the set $S \backslash \{a_1, \dots, a_{n-2}\}$. Let v be this element. In other words, in every tree T such that $f(T) = (a_1, \dots, a_{n-2})$ the vertex v is a leaf whose unique neighbour is a_1.

By induction, there is a unique tree T' with vertex set $S \backslash v$ such that $f(T') = (a_2, \dots, a_{n-2})$. Adding the vertex v and the edge (a_1, v) to T' yields the desired unique tree T with $f(T) = (a_1, \dots, a_{n-2})$. $\square$

Example 2.15. We use the idea of the above proof to compute the tree with Prüfer code (1,6,6,3,1).

(1,6,6,3,1)
2 is the smallest leaf,
incident to $a_1 = 1$

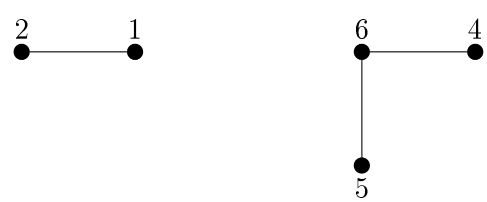


$\{1, 3, 4, 5, 6, 7\}$ left

(6,6,3,1)
4 is the smallest leaf
incident to 6

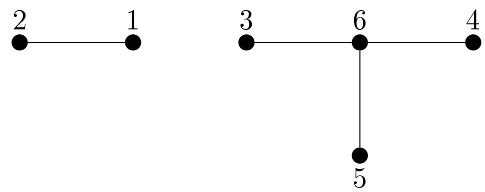
$\{1, 3, 5, 6, 7\}$ left

(6,3,1)
5 is the smallest leaf,
incident to 6



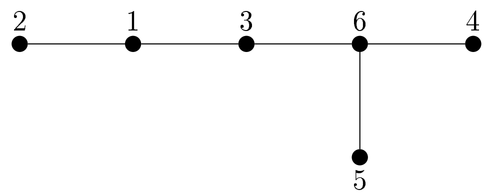
$\{1, 3, 6, 7\}$ left

(3,1)
6 is the smallest leaf
incident to 3



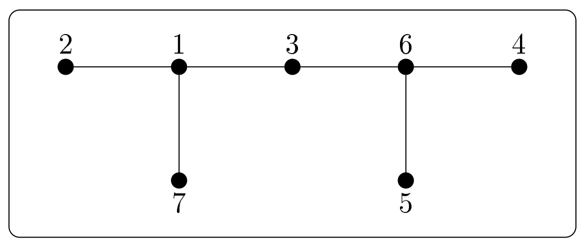
$\{1, 3, 7\}$ left

(1)
3 is the smallest leaf
incident to 1



$\{1, 7\}$ left

Now add an edge between
the remaining vertices $\{1, 7\}$.



To prove Cayley's formula, just apply Proposition 2.14 with the vertex set $[n]$ (note that there are n^{n-2} sequences in $[n]^{n-2}$).

2.4 Cayley's formula — directed graphs

For our second proof of Cayley's formula we need the following definition.

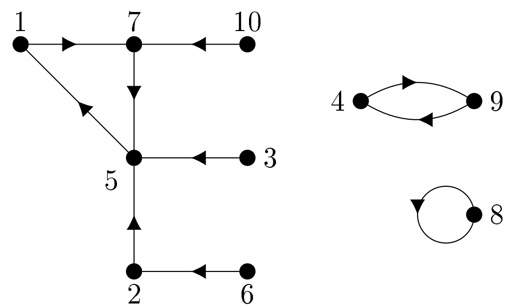
Definition 2.16. A *directed graph*, or *digraph* for short, is a vertex set and an edge (multi-)set of *ordered* pairs of vertices. Equivalently, a digraph is a (possibly not simple) graph where each edge is assigned a direction. The *out-degree* (respectively *in-degree*) of a vertex is the number of edges incident to that vertex which point away from it (respectively, towards it).

Proof of Cayley's formula (due to Joyal, 1981). We count trees on n vertices which have two distinguished vertices called the "left end" L and the "right end" R, where L and R can be the same vertex. Let t_n be the number of labelled trees on n vertices, and let T_n be the family of labelled trees with two distinguished vertices L and R. Clearly, $|T_n| = t_n n^2$, and it is thus enough to prove that $|T_n| = n^n$. We'll describe a bijection between T_n and the set of all mappings $f : [n] \to [n]$. As the number of such mappings is clearly n^n, the result will follow.

So, let $f : [n] \to [n]$ be a mapping. We represent f as a directed graph G_f with vertex set $[n]$ and the set of directed edges $E(G_f) = \{(i, f(i)) : 1 \leq i \leq n\}$.

Example.

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 7 & 5 & 5 & 9 & 1 & 2 & 5 & 8 & 4 & 7 \end{pmatrix},$$

$G_f =$ 

Observe that G_f is a digraph in which the outdegree of every vertex is exactly one ($f(i)$ is the only out-neighbour of i).

Let us look at a component of G_f (ignoring edge directions for a moment). Since the out-degree of every vertex is exactly one, each such component contains as many edges as vertices and has therefore exactly one cycle (by Corollary 2.8). This is easily seen to be a directed cycle (just follow an edge leaving a current vertex until you hit a previously visited vertex).

Let M be the union of the vertex sets of these cycles. In order to create a tree, we need to get rid of these cycles. It is easy to see that f restricted to M is a bijection; moreover, M is the unique maximal set on which f acts as a bijection.

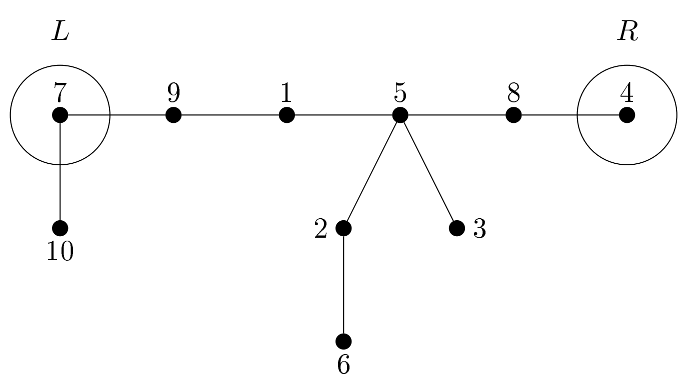
Let us write

$$f_M = \begin{pmatrix} v_1 & \dots & v_k \\ f(v_1) & \dots & f(v_k) \end{pmatrix},$$

where $v_1 < v_2 < \cdots < v_k$ (and $M = \{v_1, v_2, \dots, v_k\}$). This gives us the ordered k-tuple $(f(v_1), \dots, f(v_k))$. Now we can choose $L = f(v_1)$, $R = f(v_k)$. The tree T corresponding to f is constructed as follows: Draw a (directed) path $f(v_1), f(v_2), \dots, f(v_k)$, and fill in the remaining vertices as in G_f (removing edge directions).

Example (continued)**.**

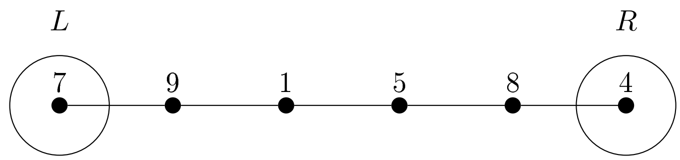
$$M = \{1, 4, 5, 7, 8, 9\},$$
$$f|_M = \begin{pmatrix} 1 & 4 & 5 & 7 & 8 & 9 \\ 7 & 9 & 1 & 5 & 8 & 4 \end{pmatrix},$$

$T = T(f) =$ 

Reversing the correspondence is easy: given a tree T with two special vertices L and R, look at the unique path P of T connecting L and R. The vertices of P form the set M. Ordering the vertices of M gives us the first line of $f|_M$, the second line is given by the order of the vertices in P, from L to R.

Example (continued)**.**

$P =$



$$M = \{1, 4, 5, 7, 8, 9\},$$
$$f|_M = \begin{pmatrix} 1 & 4 & 5 & 7 & 8 & 9 \\ 7 & 9 & 1 & 5 & 8 & 4 \end{pmatrix}.$$

The remaining values of f are then filled in accordance with the unique paths from the remaining vertices to P (directing these paths towards P). $\square$