WIKIPEDIA

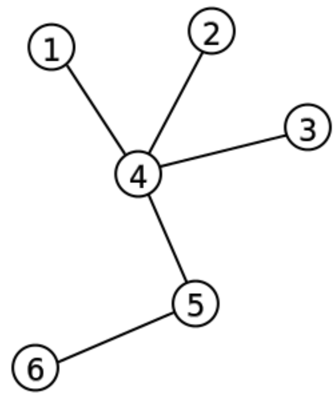
Tree (graph theory)

In mathematics, and more specifically in graph theory, a **tree** is an undirected graph in which any two vertices are connected by *exactly one* path. In other words, any acyclic connected graph is a tree. A **forest** is a disjoint union of trees.

The various kinds of data structures referred to as trees in computer science have underlying graphs that are trees in graph theory, although such data structures are generally **rooted trees**. A rooted tree may be directed, called a **directed rooted tree**,[1][2] either making all its edges point away from the root—in which case it is called an **arborescence**,[3] **branching**,[4] or **out-tree**[4]—or making all its edges point towards the root—in which case it is called an **anti-arborescence**[5] or **in-tree**.[6] A rooted tree itself has been defined by some authors as a directed graph.[7][8][9]

The term "tree" was coined in 1857 by the British mathematician Arthur Cayley.[10]

Trees



A labeled tree with 6 vertices and 5 edges.

Vertices	*v*
Edges	*v* − 1
Chromatic number	2 if *v* > 1

Contents

Definitions

Tree

A *tree* is an undirected graph *G* that satisfies any of the following equivalent conditions:

- *G* is connected and has no cycles.
- *G* is acyclic, and a simple cycle is formed if any edge is added to *G*.
- *G* is connected, but is not connected if any single edge is removed from *G*.

- G is connected and the 3-vertex complete graph K_3 is not a minor of G.
- Any two vertices in G can be connected by a unique simple path.

If G has finitely many vertices, say n of them, then the above statements are also equivalent to any of the following conditions:

- G is connected and has $n − 1$ edges.
- G has no simple cycles and has $n − 1$ edges.

As elsewhere in graph theory, the order-zero graph (graph with no vertices) is generally excluded from consideration: while it is vacuously connected as a graph (any two vertices can be connected by a path), it is not 0-connected (or even (−1)-connected) in algebraic topology, unlike non-empty trees, and violates the "one more vertex than edges" relation.

An **internal vertex** (or **inner vertex** or **branch vertex**) is a vertex of degree at least 2. Similarly, an **external vertex** (or *outer vertex*, *terminal vertex* or *leaf*) is a vertex of degree 1.

An *irreducible tree* (or *series-reduced tree*) is a tree in which there is no vertex of degree 2.

Forest

A *forest* is an undirected graph, all of whose connected components are trees; in other words, the graph consists of a disjoint union of trees. Equivalently, a forest is an undirected acyclic graph. As special cases, an empty graph, a single tree, and the discrete graph on a set of vertices (that is, the graph with these vertices that has no edges), are examples of forests. Since for every tree $V - E = 1$, we can easily count the number of trees that are within a forest by subtracting the difference between total vertices and total edges. $TV - TE$ = number of trees in a forest.

Polytree

A *polytree*[11] (or *oriented tree*[12][13] or *singly connected network*[14]) is a directed acyclic graph (DAG) whose underlying undirected graph is a tree. In other words, if we replace its directed edges with undirected edges, we obtain an undirected graph that is both connected and acyclic.

A *directed tree* is a directed graph which would be a tree if the directions on the edges were ignored, i.e. a polytree. Some authors restrict the phrase to the case where the edges are all directed towards a particular vertex, or all directed away from a particular vertex (see arborescence).

Rooted tree

A *rooted tree* is a tree in which one vertex has been designated the *root*. The edges of a rooted tree can be assigned a natural orientation, either *away from* or *towards* the root, in which case the structure becomes a *directed rooted tree*. When a directed rooted tree has an orientation away from the root, it is called an *arborescence*, *branching*, or *out-tree*; when it has an orientation towards the root, it is called an *anti-arborescence* or *in-tree*. The *tree-order* is the partial ordering on the vertices of a tree with $u < v$ if and only if the unique path from the root to v passes through u. A rooted tree which is a subgraph of some graph G is a normal tree if the ends of every edge in G are comparable in this tree-order whenever those ends are vertices of the tree (Diestel 2005, p. 15). Rooted trees, often with additional structure such as ordering of the neighbors at each vertex, are a key data structure in computer science; see tree data structure.

In a context where trees are supposed to have a root, a tree without any designated root is called a *free tree*.

A *labeled tree* is a tree in which each vertex is given a unique label. The vertices of a labeled tree on n vertices are typically given the labels 1, 2, …, n. A *recursive tree* is a labeled rooted tree where the vertex labels respect the tree order (i.e., if $u < v$ for two vertices u and v, then the label of u is smaller than the label of v).

In a rooted tree, the *parent* of a vertex is the vertex connected to it on the path to the root; every vertex except the root has a unique parent. A *child* of a vertex *v* is a vertex of which *v* is the parent. A *descendant* of any vertex *v* is any vertex which is either the child of *v* or is (recursively) the descendant of any of the children of *v*. A *sibling* to a vertex *v* is any other vertex on the tree which has the same parent as *v*. The root is an external vertex if it has precisely one child. A leaf is different from the root.

The *height* of a vertex in a rooted tree is the length of the longest downward path to a leaf from that vertex. The *height* of the tree is the height of the root. The *depth* of a vertex is the length of the path to its root (*root path*). This is commonly needed in the manipulation of the various self-balancing trees, AVL trees in particular. The root has depth zero, leaves have height zero, and a tree with only a single vertex (hence both a root and leaf) has depth and height zero. Conventionally, an empty tree (tree with no vertices, if such are allowed) has depth and height −1.

A *k-ary tree* is a rooted tree in which each vertex has at most *k* children.[15] 2-ary trees are often called *binary trees*, while 3-ary trees are sometimes called *ternary trees*.

Ordered tree

An *ordered tree* (or *plane tree*) is a rooted tree in which an ordering is specified for the children of each vertex. This is called a "plane tree" because an ordering of the children is equivalent to an embedding of the tree in the plane, with the root at the top and the children of each vertex lower than that vertex. Given an embedding of a rooted tree in the plane, if one fixes a direction of children, say left to right, then an embedding gives an ordering of the children. Conversely, given an ordered tree, and conventionally drawing the root at the top, then the child vertices in an ordered tree can be drawn left-to-right, yielding an essentially unique planar embedding .

Properties

- Every tree is a bipartite graph and a median graph. Every tree with only countably many vertices is a planar graph.
- Every connected graph *G* admits a spanning tree, which is a tree that contains every vertex of *G* and whose edges are edges of *G*.
- Every connected graph with only countably many vertices admits a normal spanning tree (Diestel 2005, Prop. 8.2.4).
- There exist connected graphs with uncountably many vertices which do not admit a normal spanning tree (Diestel 2005, Prop. 8.5.2).
- Every finite tree with *n* vertices, with *n* > 1, has at least two terminal vertices (leaves). This minimal number of leaves is characteristic of path graphs; the maximal number, *n* − 1, is attained only by star graphs. The number of leaves is at least the maximal vertex degree.
- For any three vertices in a tree, the three paths between them have exactly one vertex in common.
- Otter showed that any n-vertex tree has either a unique center vertex, whose removal splits the tree into subtrees of fewer than n/2 vertices, or a unique center edge, whose removal splits the tree into two subtrees of exactly n/2 vertices.

Enumeration

Labeled trees

Cayley's formula states that there are n^{n-2} trees on *n* labeled vertices. A classic proof uses Prüfer sequences, which naturally show a stronger result: the number of trees with vertices 1, 2, …, *n* of degrees $d_1, d_2, \ldots, d_n$ respectively, is the multinomial coefficient

$$\binom{n-2}{d_1-1, d_2-1, \ldots, d_n-1}.$$

A more general problem is to count spanning trees in an undirected graph, which is addressed by the matrix tree theorem. (Cayley's formula is the special case of spanning trees in a complete graph.) The similar problem of counting all the subtrees regardless of size has been shown to be #P-complete in the general case (Jerrum (1994)).

Unlabeled trees

Counting the number of unlabeled free trees is a harder problem. No closed formula for the number $t(n)$ of trees with n vertices up to graph isomorphism is known. The first few values of $t(n)$ are

> 1, 1, 1, 1, 2, 3, 6, 11, 23, 47, 106, 235, 551, 1301, 3159, … (sequence A000055 in the OEIS).

Otter (1948) proved the asymptotic estimate

$$t(n) \sim C\alpha^n n^{-5/2} \quad \text{as } n \to \infty,$$

with the values C and α known to be approximately 0.534949606… and 2.95576528565… (sequence A051491 in the OEIS), respectively. (Here, $f \sim g$ means that $\lim_{n\to\infty} f/g = 1$.) This is a consequence of his asymptotic estimate for the number $r(n)$ of unlabeled rooted trees with n vertices:

$$r(n) \sim D\alpha^n n^{-3/2} \quad \text{as } n \to \infty,$$

with D around 0.43992401257… and the same α as above (cf. Knuth (1997), chap. 2.3.4.4 and Flajolet & Sedgewick (2009), chap. VII.5, p. 475).

The first few values of $r(n)$ are[16]

> 1, 1, 2, 4, 9, 20, 48, 115, 286, 719, 1842, 4766, 12486, 32973, …

Types of trees

- A *path graph* (or *linear graph*) consists of n vertices arranged in a line, so that vertices i and i+1 are connected by an edge for i=1,…,n−1.
- A *starlike tree* consists of a central vertex called *root* and several path graphs attached to it. More formally, a tree is starlike if it has exactly one vertex of degree greater than 2.
- A *star tree* is a tree which consists of a single internal vertex (and n−1 leaves). In other words, a star tree of order n is a tree of order n with as many leaves as possible.
- A *caterpillar tree* is a tree in which all vertices are within distance 1 of a central path subgraph.
- A *lobster tree* is a tree in which all vertices are within distance 2 of a central path subgraph.

See also

- Hypertree
- Tree structure
- Tree (data structure)
- Decision tree
- Pseudoforest
- Unrooted binary tree

Notes

1. Stanley Gill Williamson (1985). *Combinatorics for Computer Science*. Courier Dover Publications. p. 288. ISBN 978-0-486-42076-9.

2. Mehran Mesbahi; Magnus Egerstedt (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press. p. 38. ISBN 1-4008-3535-6.
3. Ding-Zhu Du; Ker-I Ko; Xiaodong Hu (2011). *Design and Analysis of Approximation Algorithms*. Springer Science & Business Media. p. 108. ISBN 978-1-4614-1701-9.
4. Jonathan L. Gross; Jay Yellen; Ping Zhang (2013). *Handbook of Graph Theory, Second Edition*. CRC Press. p. 116. ISBN 978-1-4398-8018-0.
5. Bernhard Korte; Jens Vygen (2012). *Combinatorial Optimization: Theory and Algorithms* (5th ed.). Springer Science & Business Media. p. 28. ISBN 978-3-642-24488-9.
6. Kurt Mehlhorn; Peter Sanders (2008). *Algorithms and Data Structures: The Basic Toolbox* (http://people.mpi-inf.mpg.de/~mehlhorn/ftp/Toolbox/Introduction.pdf) (PDF). Springer Science & Business Media. p. 52. ISBN 978-3-540-77978-0.
7. David Makinson (2012). *Sets, Logic and Maths for Computing*. Springer Science & Business Media. pp. 167–168. ISBN 978-1-4471-2499-3.
8. Kenneth Rosen (2011). *Discrete Mathematics and Its Applications, 7th edition*. McGraw-Hill Science. p. 747. ISBN 978-0-07-338309-5.
9. Alexander Schrijver (2003). *Combinatorial Optimization: Polyhedra and Efficiency*. Springer. p. 34. ISBN 3-540-44389-4.
10. Cayley (1857) "On the theory of the analytical forms called trees," (https://books.google.com/books?id=MlEEAAAAYAAJ&pg=PA172#v=onepage&q&f=false) *Philosophical Magazine*, 4th series, **13** : 172–176.
However it should be mentioned that in 1847, K.G.C. von Staudt, in his book *Geometrie der Lage* (Nürnberg, (Germany): Bauer und Raspe, 1847), presented a proof of Euler's polyhedron theorem which relies on trees on pages 20–21 (https://books.google.com/books?id=MzQAAAAAQAAJ&pg=PA20#v=onepage&q&f=false). Also in 1847, the German physicist Gustav Kirchhoff investigated electrical circuits and found a relation between the number (n) of wires/resistors (branches), the number (m) of junctions (vertices), and the number (μ) of loops (faces) in the circuit. He proved the relation via an argument relying on trees. See: Kirchhoff, G. R. (1847) "Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird" (https://books.google.com/books?id=gx4AAAAAMAAJ&vq=Kirchoff&pg=PA497#v=onepage&q&f=false) (On the solution of equations to which one is led by the investigation of the linear distribution of galvanic currents), *Annalen der Physik und Chemie*, **72** (12) : 497–508.
11. See Dasgupta (1999).
12. See Harary & Sumner (1980).
13. See Simion (1991).
14. See Kim & Pearl (1983).
15. See Black, Paul E. (4 May 2007). "k-ary tree" (http://xlinux.nist.gov/dads/HTML/karyTree.html). U.S. National Institute of Standards and Technology. Retrieved 8 February 2015.
16. See Li (1996).

References

- Dasgupta, Sanjoy (1999), "Learning polytrees", *in Proc. 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999), Stockholm, Sweden, July–August 1999* (http://cseweb.ucsd.edu/~dasgupta/papers/poly.pdf) (PDF), pp. 134–141.
- Harary, Frank; Sumner, David (1980), "The dichromatic number of an oriented tree", *Journal of Combinatorics, Information & System Sciences*, **5** (3): 184–187, MR 0603363 (https://www.ams.org/mathscinet-getitem?mr=0603363).
- Kim, Jin H.; Pearl, Judea (1983), "A computational model for causal and diagnostic reasoning in inference engines", *in Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI 1983), Karlsruhe, Germany, August 1983* (http://ijcai.org/Past%20Proceedings/IJCAI-83-VOL-1/PDF/041.pdf) (PDF), pp. 190–193.
- Li, Gang (1996), "Generation of Rooted Trees and Free Trees", *M.S. Thesis, Dept. of Computer Science, University of Victoria, BC, Canada* (http://webhome.cs.uvic.ca/~ruskey/Theses/GangLiMScThesis.pdf) (PDF), p. 9.
- Simion, Rodica (1991), "Trees with 1-factors and oriented trees", *Discrete Mathematics*, **88** (1): 93–104, doi:10.1016/0012-365X(91)90061-6 (https://doi.org/10.1016%2F0012-365X%2891%2990061-6), MR 1099270 (https://www.ams.org/mathscinet-getitem?mr=1099270).

Further reading

- Diestel, Reinhard (2005), *Graph Theory* (http://diestel-graph-theory.com/index.html) (3rd ed.), Berlin, New York: Springer-Verlag, ISBN 978-3-540-26183-4.
- Flajolet, Philippe; Sedgewick, Robert (2009), *Analytic Combinatorics*, Cambridge University Press, ISBN 978-0-521-89806-5
- Hazewinkel, Michiel, ed. (2001) [1994], "Tree" (https://www.encyclopediaofmath.org/index.php?title=p/t094060), *Encyclopedia of Mathematics*, Springer Science+Business Media B.V. / Kluwer Academic Publishers, ISBN 978-1-55608-010-4
- Knuth, Donald E. (November 14, 1997), *The Art of Computer Programming Volume 1: Fundamental Algorithms* (3rd ed.), Addison-Wesley Professional
- Jerrum, Mark (1994), "Counting trees in a graph is #P-complete", *Information Processing Letters*, **51** (3): 111–116, doi:10.1016/0020-0190(94)00085-9 (https://doi.org/10.1016%2F0020-0190%2894%2900085-9), ISSN 0020-0190 (https://www.worldcat.org/issn/0020-0190).
- Otter, Richard (1948), "The Number of Trees", *Annals of Mathematics*, Second Series, **49** (3): 583–599, doi:10.2307/1969046 (https://doi.org/10.2307%2F1969046), JSTOR 1969046 (https://www.jstor.org/stable/1969046).