Graphs and Trees

Robert Y. Lewis

CS 0220 2023

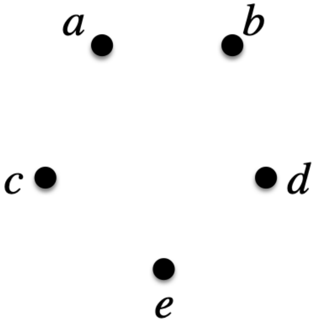
April 21, 2023

Overview

Empty graph

Let G be a graph. Often convenient to let $n = |V(G)|$, the number of vertices, and $m = |E(G)|$, the number of edges.

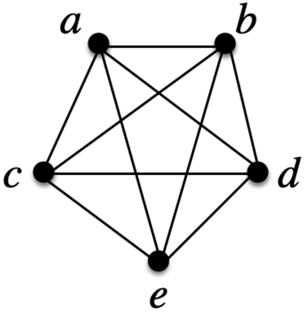
We insist that $n > 0$. But, if $m = 0$, we call it the *empty graph* on n nodes.



Complete graph

If, $\forall u, v \in V(G), u \neq v \rightarrow \{u, v\} \in E(G)$, we say G is a *complete graph* (or a clique). How do we read that in English? There is an edge between every two distinct vertices. Also known as K_n.

$m = \binom{n}{2} = n(n-1)/2$.

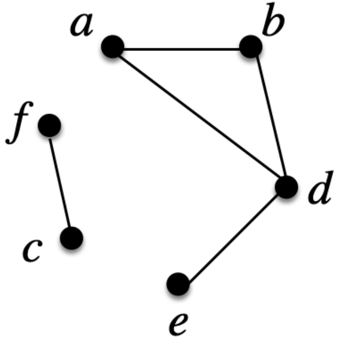


Path

A length-k path p in G is

- an element of $V(G)^{k+1}$
- such that $\forall i \in [1, k], \{p^i, p^{i+1}\} \in E(G)$.

The path is *simple* if there is no $i \neq i'$ such that $p^i = p^{i'}$.

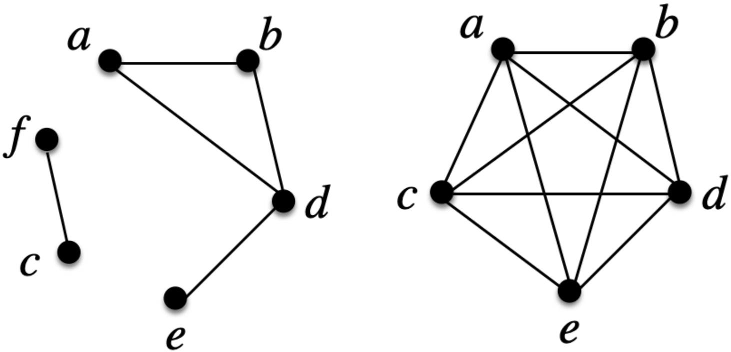


Paths? (a, b, d, a)? Yes (not simple). (d, e, a)? No, missing edge.

Connected graph

A pair of vertices u and v is *connected* if there exists a length-k path such that $p^1 = u$ and $p^{k+1} = v$.

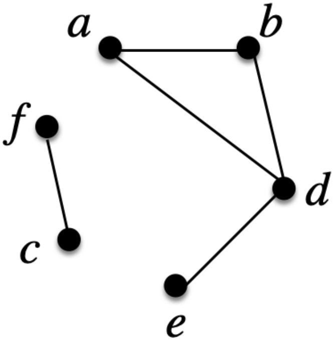
A graph G is connected if all pairs of vertices in $V(G)$ are connected: for all $u, v \in V(G)$, there is a (u, v)-path.



Cycle

A length-k cycle p in G is

- a length-k path in G
- where $p^{k+1} = p^1$.



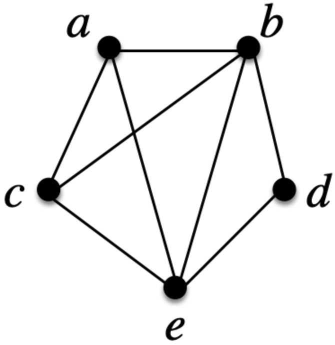
Cycles? (a, b, d, a)? Yes. (b, a, e, b)? No, missing edge. (e, d, b, d, e)? Yes. (f, c, f)? Yes.

Edge and vertex sets of paths

For a length-k path p, let $E(p) = \{\{p^i, p^{i+1}\} \mid i \in [1, k]\}$.

Let $V(p) = \{p^i \mid i \in [1, k+1]\}$. These are p's edge and vertex sets.

Find two cycles p and q such that $V(p) = V(q)$ but $E(p) \neq E(q)$.

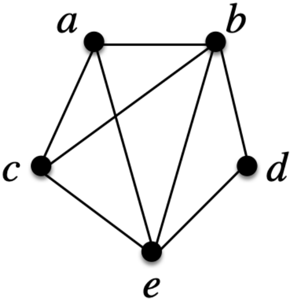


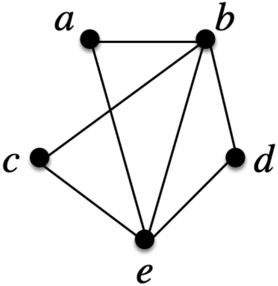
Tours on Graphs

Let G be a graph with n vertices and m edges. Let p be a length-k cycle in G.

Definition: p is an *Eulerian tour* (edge tour) of G if $E(p) = E(G)$ and $k = m$.

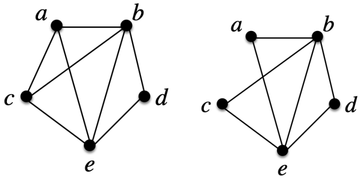
Definition: p is a *Hamiltonian tour* (vertex tour) of G if $V(p) = V(G)$ and $k = n$.





Eulerian implies even

If a graph has an Eulerian tour, deg(v) must be even for all vertices.

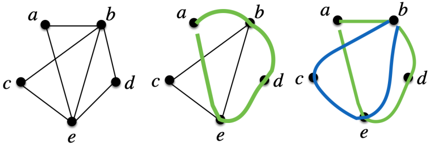


Proof sketch: Let p be an Eulerian tour of G. Every time the tour visits a vertex, it must arrive and depart (on fresh edges). Thus, each appearence of v on the cycle accounts for two edges that include v. In total, an even number.

Because it's an Eulerian cycle, every edge of the graph appears exactly once. Thus, every vertex of the graph is adjacent to an even number of other vertices.

Even implies Eulerian

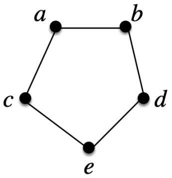
If all vertices in a connected graph have even degree, it has an Eulerian path (!).



Pick a vertex. Pick an arbitrary unused edge (if there is one) to go to another vertex. This process can only get stuck when we've returned to the starting vertex. That's the only vertex with an odd number of unused edges. If not an Eulerian tour, there's some vertex that we visited that has unused edges. That's because the graph is connected. Pick such a vertex and start this process again, rerouting our original route through this side route. By the earlier argument, the side route has to return to its starting point. Process must complete and can only complete with an Eulerian tour.

Cycle graph

If an n-node graph G has a cycle p that is both an Eulerian tour *and* a Hamiltonian tour, we call the graph a cycle and write it C_n.



Equivalent definition: C_n is connected and $\forall v \in V(G), \deg(v) = 2$.

Number of edges of C_n is n.

For what values of n, if any, does $K_n = C_n$?

Puzzles

- When does C_n have an Eulerian/Hamiltonian tour?
- When does K_n have an Eulerian/Hamiltonian tour?
- How many simple paths in K_n of length $n - 1$?
- How many simple paths in C_n of length $n - 1$?

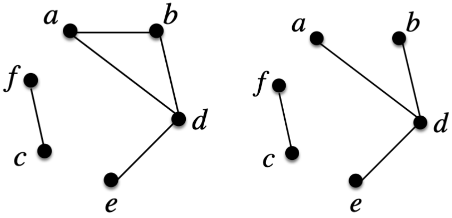
Acyclic graph

Cycle is a path p where the first and last vertices are the same.

Definition: A *simple cycle* is a length $k \geq 3$ cycle with no repeated vertices (except the first and last).

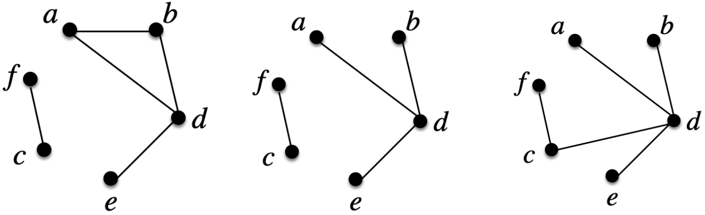
Definition: A graph is *acyclic* if it contains no simple cycles.

In an acyclic graph, if there is a simple path from u to v, there is only one such path. If there were more than one such path, we could use it to build a simple cycle.



Tree

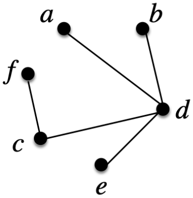
Definition: A *tree* is a connected acyclic graph.



Define the "connected to" relation in a graph G on a pair of vertices u and v as whether there is a path from u to v or $u = v$. Symmetric? Yes. Reflexive? Yes. Transitive? Yes. Equivalence relation! Thus, "connected to" partitions the graph into "connected components".

Since an acyclic graph can be decomposed into a set of trees (connected, acyclic), it can be called a *forest*.

Leaves

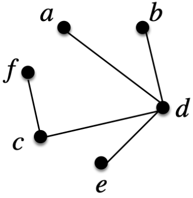


Definition: A *leaf* is a vertex v in a tree such that $\deg(v) = 1$.

An n-vertex star graph is a tree with one central vertex and $n - 1$ leaves. An n-vertex chain is a simple path with 2 leaves.

Depth and parents

We can define any vertex of a tree to be its root.



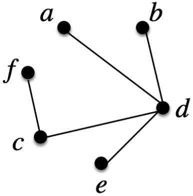
Definition: Given a tree G and a choice of root $r \in V(G)$, the *depth* of $u \in V(G)$, $\text{dep}_r(u)$ is the length of the simple path from r to u.

Depth is well defined because every pair of nodes in a tree has a unique simple path between them.

Definition: Given a tree G and a choice of root $r \in V(G)$, u is the *parent* of v if $(u, v) \in E(G)$ and $\text{dep}_r(u) = \text{dep}_r(v) - 1$.

All my children

In a tree G with root r, if $(u, v) \in E(G)$, $|\text{dep}_r(u) - \text{dep}_r(v)| = 1$. Proof?



In a tree, every vertex (except the root) has exactly one parent. Proof?

Definition: If u is the parent of v, we call v a *child* of u.

A (non-root) leaf has no children. Other vertices have one or more children. Since v has a unique parent, it has $\deg(v) - 1$ children.

Subgraph

Let G be a graph. A subgraph G' of G is defined so that

- $V(G') \subseteq V(G)$
- $E(G') \subseteq E(G)$
- $\forall (u, v) \in E(G'), u \in V(G') \wedge v \in V(G')$

Example facts:

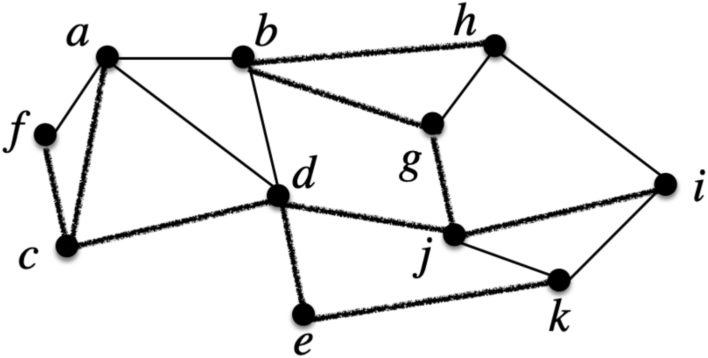
- If u is connected to v in G', then u is connected to v in G.
- All n-node graphs are subgraphs of a complete graph K_n.
- Every subgraph of an acyclic graph is acyclic.

Properties of trees

1. *Every connected subgraph is a tree.* Proof by contradiction. If subgraph is not a tree, it has simple cycles. But then so must the original tree.
2. *There is a unique path between every pair of vertices.* Otherwise, we can make a simple cycle.
3. *Adding an edge between nonadjacent vertices in a tree creates a graph with a simple cycle.* All pairs of vertices have a unique (simple) path, so adding that edge makes a simple cycle.
4. *Removing any edge disconnects the graph.* There is no longer a path between the endpoints of the edge.
5. *If the tree has at least two vertices, then it has at least two leaves.* Chains are trees and have the smallest number of leaves.
6. *The number of vertices in a tree is one larger than the number of edges.* Can prove by induction.

Definition

Definition: A *spanning tree* of a graph G is a subgraph of G that (1) is a tree and (2) includes all of the vertices of G.



Existence of spanning trees

Theorem: Every connected graph contains a spanning tree.

Proof: G is a subgraph of G that is connected and includes all of the vertices of G. It has $m = |E(G)|$ edges. By the well-ordering principle, there must be a *smallest* graph T with this property.

T must be a spanning tree. Since T is a connected graph that includes all of the vertices of G, all we have to show is that T is acyclic.

Suppose to the contrary that T contains a simple cycle C. Removing any edge of the cycle results in a graph T' that still includes all of the vertices of G and is still connected. But that violates the definition of T. Therefore, G contains a spanning tree.