Euler Circuits and Trails

Table of contents

Euler circuits 1
Euler trails 2
The theorem 2
This course's definition and the textbook's 6
How to find one 7
Deciding: the flowchart 8
The bridges of Königsberg 11
Common mistakes 11
See also 11

An Euler circuit goes along every edge of a graph exactly once and comes back to where it started. An Euler trail does the same but finishes somewhere else. Whether a graph has one is decided by two things only: whether it is connected once its isolated vertices are set aside, and how many of its vertices have odd degree. None gives an Euler circuit, exactly two gives an Euler trail from one of them to the other, and any other number gives neither. This note has the definitions, the theorem with a full proof, how to actually find a circuit or trail, a flowchart for deciding, and how this course's definition differs from the textbook's. It builds on walks, trails, circuits and connectedness in walks-trails-and-circuits[1], and on degree in graphs[2]. First met in 2026-10-26-walks-trails-and-circuits[3].

Euler circuits

An **Euler circuit** is a circuit that contains every edge of the graph.

[1] courses/math1061/walks-trails-and-circuits.pdf
[2] courses/math1061/graphs.pdf
[3] courses/math1061/2026-10-26-walks-trails-and-circuits.pdf

A circuit never uses an edge twice (walks-trails-and-circuits[4]), so an Euler circuit uses **every edge exactly once** and ends where it started. It may pass through a vertex many times. A graph that has an Euler circuit is called **Eulerian**.

A graph with no edges has an Euler circuit: the trivial circuit at any vertex contains every edge, since there are none.

Euler trails

> An **Euler trail from vertex u to vertex v** is a trail that starts at u and ends at v and contains every edge of the graph.

The difference from an Euler circuit is only where it finishes. Both use every edge exactly once. An Euler circuit ends at the vertex it started from. An Euler trail from u to v ends at v, and the case that matters is $u \neq v$, an open route. (An Euler circuit is a trail from a vertex to itself, but the theorem below is about $u \neq v$.)

For a connected graph the two cannot both happen with $u \neq v$: the theorem says an Euler circuit needs every degree even, and an Euler trail from u to $v \neq u$ needs u and v odd.

The theorem

> **Theorem.** Let G be a connected graph. Then G has an Euler circuit if and only if every vertex of G has even degree.
>
> Let $u, v \in V(G)$ be distinct vertices of G. Then G has an Euler trail from u to v if and only if u and v have odd degree and all other vertices of G have even degree.

Every graph has an even number of vertices of odd degree (graphs[5]), so for a connected graph there are three cases: no odd vertices gives an Euler circuit, exactly two gives an Euler trail between them, and four or more gives neither.

Counting edge ends along a trail

Both directions of the proof rest on one count. Take a trail $T = v_0\, e_1\, v_1 \cdots e_n\, v_n$ and a vertex w. The edge e_i has its two ends at v_{i-1} and v_i (both at the same vertex if it is a loop). So the number of edge ends at w that T uses is

$$\underbrace{\#\{\, i : v_{i-1} = w \,\}}_{\text{times } T \text{ leaves } w} + \underbrace{\#\{\, i : v_i = w \,\}}_{\text{times } T \text{ arrives at } w}.$$

[4] courses/math1061/walks-trails-and-circuits.pdf
[5] courses/math1061/graphs.pdf

These ends are all different, because the edges of a trail are all different. Reading along the sequence, each arrival at w is followed by a departure, except an arrival at the very end, and each departure is preceded by an arrival, except a departure at the very start. So:

- **If w is not the first or last vertex**, T arrives at w exactly as often as it leaves: every pass through w uses **two** ends, one in and one out, and the number used is even.
- **If T is a circuit** ($v_0 = v_n$), the departure at the start and the arrival at the end pair up at v_0, and every vertex has an even number of ends used.
- **If** $v_0 \neq v_n$, T leaves v_0 once more than it arrives, and arrives at v_n once more than it leaves. At those two vertices the number of ends used is **odd**.

A loop at w is an arrival at w and a departure from w at once, and uses two ends at w, so it never changes these parities.

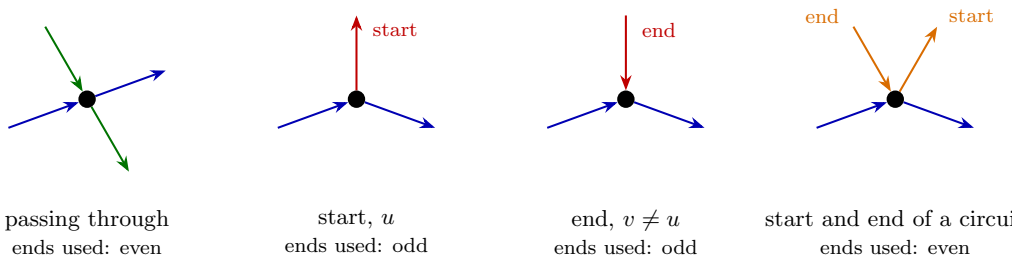


Figure 1: Edge ends used by a trail at one vertex. Arrows show the direction of travel. Left: a vertex in the middle of the trail. Each time the trail passes through, it arrives along one edge and leaves along another, so the ends pair up, blue with blue and green with green, and an even number is used. Second: the start of a trail that ends elsewhere has one departure with no arrival to pair with, in red, so an odd number is used. Third: the end of such a trail has one arrival left over. Right: at the start of a circuit, the first departure pairs with the final arrival, so every end pairs up again.

Proof: every edge used forces the degrees

Proof that an Euler circuit or trail forces the degrees. Suppose T is an Euler circuit of G. It uses every edge exactly once, so it uses every edge end in G exactly once, and the number of ends it uses at a vertex w is all of them, $\deg(w)$. By the count above that number is even. So every vertex has even degree.

Suppose instead T is an Euler trail from u to v with $u \neq v$. Again the ends used at w number $\deg(w)$. By the count, that is odd for $w = u$ and $w = v$, and even for every other vertex. □

This direction does not use connectedness.

Proof: even degrees give an Euler circuit

The construction rests on one fact about walking until stuck.

Lemma (stuck only at the start). Let H be a graph in which every vertex has even degree, and w a vertex with $\deg(w) \geq 1$. Start at w and build a trail one edge at a time: from the current vertex, go along any edge at it that has not been used yet, and stop when there is none. Then this stops, and it stops at w, so it produces a non-trivial circuit through w.

Proof. Each step uses a new edge and there are finitely many, so it stops. Suppose it is at a vertex $x \neq w$, having arrived there. The trail so far starts at w and ends at $x \neq w$, so by the count it has used an odd number of the ends at x. Since $\deg(x)$ is even, some end at x is unused, and the edge it belongs to is unused, because an edge is used all at once. So the trail can go on, and it does not stop at x. It therefore stops at w. It started along an edge, since $\deg(w) \geq 1$, so the circuit has at least one edge. □

Removing a circuit keeps every degree even. If C is a circuit and its edges are deleted (keeping all the vertices), each vertex loses the ends of it that C used, an even number by the count. Even minus even is even.

Proof of the theorem, circuits. Let G be connected with every vertex of even degree. Every vertex is a trivial circuit, so G has circuits. A circuit uses each edge at most once, so it has at most $|E(G)|$ edges, and there is a circuit $C = v_0\, e_1\, v_1 \cdots e_n\, v_n$ with **as many edges as possible**. Suppose, for a contradiction, that some edge of G is not in C.

Let H be G with the edges of C deleted. H has every vertex of even degree and at least one edge.

Some vertex of C has an edge in H. Take an edge of H and an endpoint x of it. If x is on C, done. If not, G is connected, so there is a walk in G from x to v_0. The walk starts off C and ends on C, so there is a first vertex y on it that lies on C, and the vertex z just before y is not on C. The edge f of the walk joining z and y is not an edge of C, since every edge of C has both endpoints on C. So f is in H, and its endpoint y is on C. Call this vertex w.

Splice. w has degree at least 1 in H, so the lemma, applied in H, gives a non-trivial circuit $C' = w\, f_1\, u_1 \cdots f_m\, w$ using only edges of H. w is on C, say $w = v_k$. Replace that one occurrence of v_k in C by the whole of C':

$$v_0\, e_1\, v_1 \cdots e_k\, \underbrace{v_k\, f_1\, u_1 \cdots f_m\, v_k}_{C'}\, e_{k+1} \cdots e_n\, v_n.$$

This is a walk, since C' leaves from v_k and returns to v_k. Its edges are those of C and those of C', which are all different, since the edges of C' are in H and H has none of the edges of C. So it is a trail, and it starts and ends at $v_0 = v_n$, so it is a circuit. It has $n + m > n$ edges, contradicting the choice of C.

So C contains every edge of G, and it is an Euler circuit. □

If G has no edges, C is a trivial circuit and nothing is missing. Connectedness is used in exactly one place: to get from an unused edge back to the circuit.

Proof: two odd vertices give an Euler trail

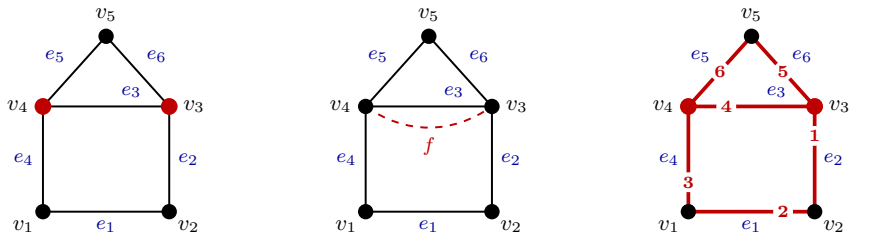
Proof of the theorem, trails. Let G be connected, with $u \neq v$ of odd degree and every other vertex of even degree. Make a new graph G' by adding one new edge f with endpoints u and v. If u and v are already adjacent, f is a parallel edge, and that is fine. In G', u and v each gain one end, so every vertex of G' has even degree, and G' is still connected. By the circuit case, G' has an Euler circuit C.

C uses f once, either as $\cdots u\, f\, v \cdots$ or as $\cdots v\, f\, u \cdots$. Read backwards if necessary (walks-trails-and-circuits[6]) so that it is $\cdots v\, f\, u \cdots$. Then start the circuit at that u, so that f is the last edge:

$$C = u \cdots v\, f\, u.$$

Delete the final $f\, u$. What is left is a trail, since its edges are some of the distinct edges of C. It starts at u, ends at v, uses every edge of G' except f exactly once, and so uses every edge of G exactly once. It is an Euler trail from u to v. □

Together with the first direction, that proves the theorem.



G: $\deg v_3 = \deg v_4 = 3$ | G': add f, all degrees even | Euler trail from v_3 to v_4

Figure 2: The trail case from the circuit case. Left: the house graph, connected, with v_3 and v_4 of odd degree, in red, and the others even. Middle: adding a new edge f joining v_3 and v_4, dashed, makes every degree even, so there is an Euler circuit, for example $v_3, e_2, v_2, e_1, v_1, e_4, v_4, e_3, v_3, e_6, v_5, e_5, v_4, f, v_3$, which ends with f. Here f is a second edge between two vertices that were already adjacent. Right: drop the final f. What is left, $v_3, e_2, v_2, e_1, v_1, e_4, v_4, e_3, v_3, e_6, v_5, e_5, v_4$, is an Euler trail from v_3 to v_4, in red with its edges numbered in order.

[6] courses/math1061/walks-trails-and-circuits.pdf

This course's definition and the textbook's

This course uses the standard definitions above: an Euler circuit contains every **edge**. The textbook defines an Euler circuit as a circuit that contains **every vertex and every edge** of the graph, and an Euler trail likewise.

They differ only when there are vertices that no edge touches. A triangle together with one isolated vertex has an Euler circuit by this course's definition (go once round the triangle), and none by the textbook's, because no circuit can reach the isolated vertex.

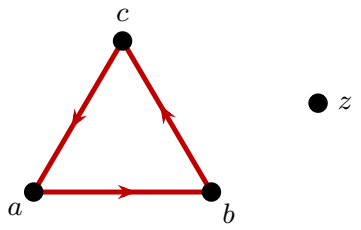


Figure 3: A triangle and an isolated vertex z. Going once round the triangle, in red, uses every edge, so by this course's definition it is an Euler circuit. By the textbook's definition it is not, since it misses z, and no circuit can visit z. The graph is disconnected, so the theorem does not apply to it as it stands. Deleting z leaves a connected graph with all degrees even.

For a connected graph the two definitions agree. A circuit or trail that contains every vertex and every edge certainly contains every edge. Conversely, let G be connected and let T contain every edge.

- If G has at least one edge, then every vertex is the endpoint of some edge. With a single vertex, every edge is a loop at it. With two or more vertices, take any vertex x and another vertex y: there is a walk from x to y, and its first edge has x as an endpoint. T uses that edge, so T passes through x.
- If G has no edges, being connected means it has one vertex. The only walk in G is that single vertex, so T is the trivial circuit there, and it contains the vertex.

So T contains every vertex too. The theorem assumes G is connected, so it says the same thing under either definition. The definitions only part ways for disconnected graphs whose edges all lie in one piece, with isolated vertices as the rest, such as the triangle and isolated vertex, and that case is what the first two steps of the flowchart deal with.

How to find one

The theorem says whether an Euler circuit or trail exists. To write one down, I use the splicing method that the proof is built on (it is usually called Hierholzer's algorithm):

1. **Check first.** Delete isolated vertices, check the rest is connected, and count the vertices of odd degree. None: look for a circuit, starting anywhere. Two, u and v: look for a trail, **starting at** u.
2. **Walk until stuck.** From the start, keep going along edges not used yet, choosing freely, until there are none at the current vertex. For a circuit this always stops back at the start (the lemma). For a trail from u it always stops at v: by the count, at any other vertex the trail has used an odd number of ends of an even-degree vertex, or, at u, an even number of ends of an odd-degree vertex, and either way an end is left.
3. **If edges are left, splice.** The unused edges give every vertex even degree, since what was walked used an even number of ends at each vertex except u and v, and an odd number at those two. Pick a vertex already on the route that still has an unused edge. Walk from it along unused edges until stuck. That is back at the same vertex. Insert this circuit into the route at that vertex.
4. **Repeat** step 3 until every edge is used.

When walking, it helps to tick off each edge on the drawing as it is used, and to write the route as a list of vertices (with edge names where there are multiple edges).

Another method avoids splicing by choosing carefully as you go (Fleury's algorithm): walk from the start, and at each step **don't burn your bridges**, that is, never use an edge whose removal would cut the unused edges into two separate pieces, unless it is the only edge left at the vertex. It gives the circuit or trail in one pass, but checking every step for a bridge is slow by hand, so I splice.

Worked: an Euler circuit by splicing

> Find an Euler circuit of the graph below: six vertices, each joined to every other vertex except one, the vertex "opposite" it. The non-adjacent pairs are v_1, v_6; v_2, v_4; v_3, v_5.

There are 12 edges, $v_1v_2, v_1v_3, v_1v_4, v_1v_5, v_2v_3, v_2v_5, v_2v_6, v_3v_4, v_3v_6, v_4v_5, v_4v_6, v_5v_6$. The graph is connected and every vertex has degree 4, so there is an Euler circuit. The graph is simple, so a route can be written as a list of vertices.

Walk until stuck. From v_1:

$$v_1\, v_2\, v_3\, v_1\, v_4\, v_5\, v_1.$$

At v_1 all four edges are now used, so this is stuck, back at the start, with 6 edges. Left over: $v_2v_5, v_2v_6, v_3v_4, v_3v_6, v_4v_6, v_5v_6$.

First splice. v_2 is on the route and still has the unused edges v_2v_5 and v_2v_6. Walk from v_2 along unused edges: $v_2\, v_5\, v_6\, v_2$, stuck at v_2. Insert it where the route passes v_2:

$$v_1\ \underbrace{v_2\, v_5\, v_6\, v_2}_{\text{inserted}}\ v_3\, v_1\, v_4\, v_5\, v_1.$$

Left over: v_3v_4, v_3v_6, v_4v_6.

Second splice. v_3 is on the route and still has unused edges. Walk from it: $v_3\, v_4\, v_6\, v_3$. Insert it where the route passes v_3:

$$v_1\, v_2\, v_5\, v_6\, v_2\ \underbrace{v_3\, v_4\, v_6\, v_3}_{\text{inserted}}\ v_1\, v_4\, v_5\, v_1.$$

That uses all 12 edges, each once, and returns to v_1: an Euler circuit.

Deciding: the flowchart

For any graph, connected or not:

Why each step is right:

- **No edges.** The trivial circuit at any vertex contains every edge, so there is an Euler circuit. This uses this course's definition: by the textbook's, a graph with two or more vertices and no edges would have none.
- **Deleting isolated vertices** changes no edge, so it does not change which routes use every edge. A route that uses every edge never needs to visit an isolated vertex.
- **Not connected, after deleting isolated vertices.** Then there are vertices x and y with no walk between them, and each has an edge, since nothing isolated is left. A circuit or trail containing both edges would contain a walk from x to y. So there is neither.
- **Connected.** Now the theorem applies. The number of odd-degree vertices is even (graphs[7]), so 0, 2, or 4 or more covers everything.

[7] courses/math1061/graphs.pdf

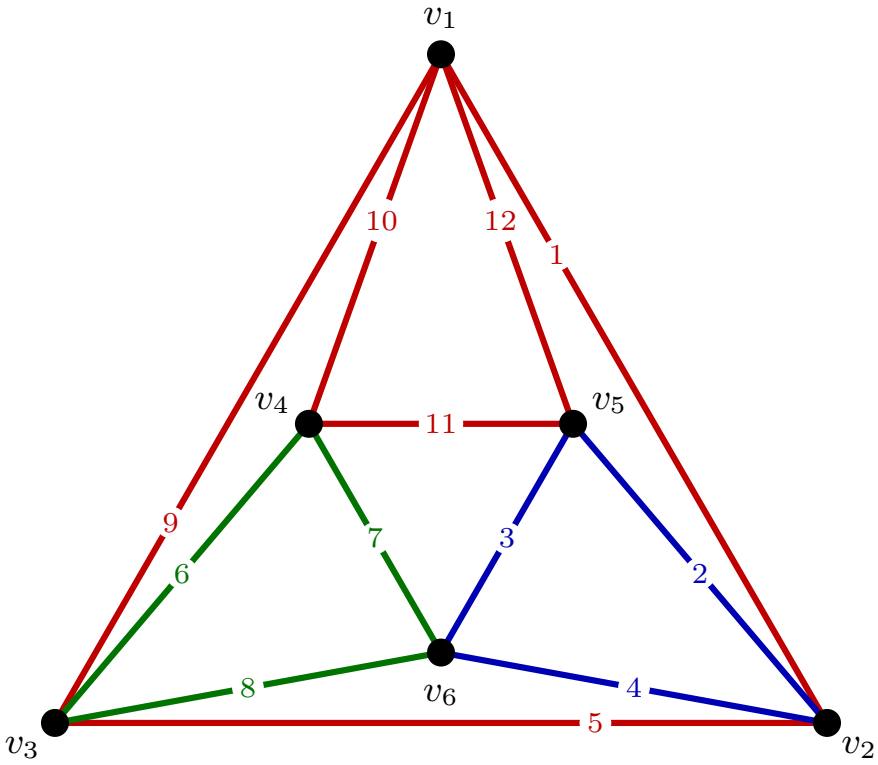


Figure 4: Finding an Euler circuit by splicing. Red: the first circuit, $v_1, v_2, v_3, v_1, v_4, v_5, v_1$, walked from v_1 until stuck. Blue: the circuit v_2, v_5, v_6, v_2 through v_2, built from unused edges and spliced in at v_2. Green: the circuit v_3, v_4, v_6, v_3 through v_3, spliced in at v_3. The small numbers give the order of the edges in the final circuit, $v_1, v_2, v_5, v_6, v_2, v_3, v_4, v_6, v_3, v_1, v_4, v_5, v_1$.

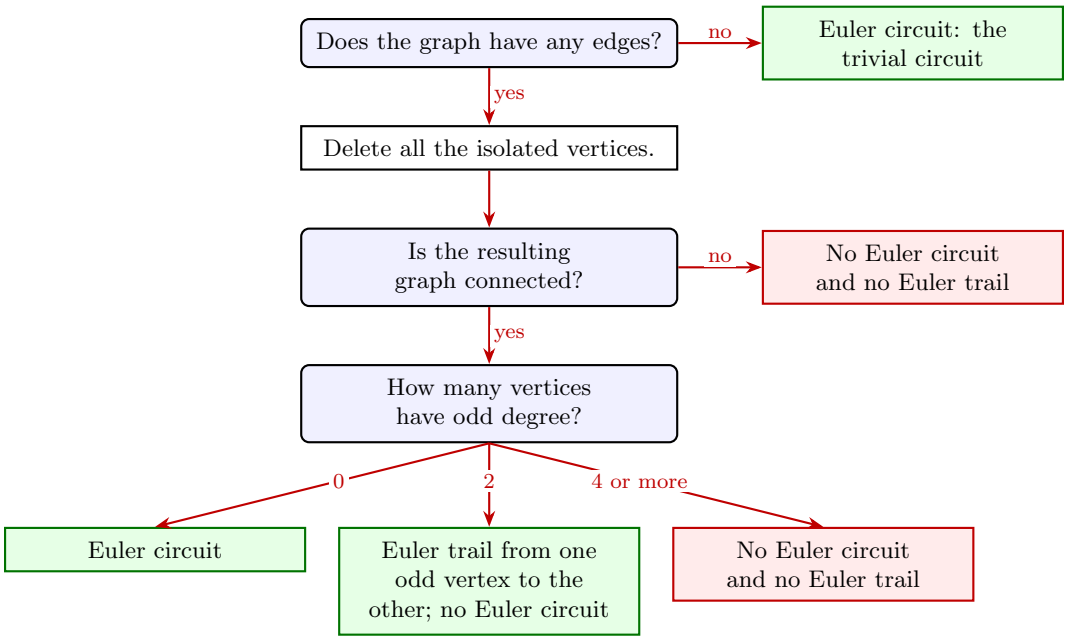


Figure 5: Deciding whether a graph has an Euler circuit or an Euler trail. A graph with no edges has one, the trivial circuit. Otherwise isolated vertices are deleted, since no edge touches them, and what is left must be connected, or some two edges cannot be on one route. Then the number of vertices of odd degree decides, and it is always even: none gives an Euler circuit, exactly two gives an Euler trail from one to the other, and four or more gives neither.

The bridges of Königsberg

The problem that started graph theory: the city of Königsberg had four land masses joined by seven bridges, and the question was whether there is a walk around the city crossing every bridge exactly once, ideally returning to the start. As a graph, the land masses are the vertices and the bridges the edges. The island has degree 5 and each of the other three land masses has degree 3 (2026-10-26-walks-trails-and-circuits[8] has the graph).

All four vertices have odd degree. The graph is connected, so by the theorem there is **no Euler circuit**, since that needs every degree even, and **no Euler trail** either, since that needs exactly two odd vertices. No walk crosses every bridge exactly once, whether or not it returns. The problem is set up in 2026-07-27-welcome-and-intro-problems[9].

Common mistakes

- **Forgetting connectedness.** Even degrees are not enough. Two separate circuits, such as two triangles, have every degree even and no Euler circuit. Check that the graph is connected, after deleting isolated vertices, before using the degrees.
- **Letting isolated vertices decide.** An isolated vertex makes the graph disconnected but does not stop an Euler circuit, by this course's definition. Delete them first. (By the textbook's definition they do stop one, which is why the definitions are compared above.)
- **Counting a loop as 1.** A loop adds 2 to the degree (graphs[10]), so loops never change whether a degree is odd or even.
- **Exactly two odd vertices gives a trail, not a circuit.** The trail must start at one odd vertex and end at the other. Starting at an even vertex will not work.
- **Zero odd vertices gives a circuit, and the trail part of the theorem does not apply.** An Euler circuit is also a trail from a vertex to itself, but the trail statement in the theorem is about $u \neq v$. With every degree even there is no Euler trail from u to a **different** v, since that would need u and v odd.
- **Getting stuck and concluding there is none.** Walking until stuck can leave edges unused even when an Euler circuit exists. That is not a failure: splice in the rest (how to find one). Only the degrees and connectedness decide existence.

See also

- walks-trails-and-circuits[11] — walks, trails, circuits and connected graphs
- graphs[12] — degree, loops and the Handshake Theorem

[8] courses/math1061/2026-10-26-walks-trails-and-circuits.pdf
[9] courses/math1061/2026-07-27-welcome-and-intro-problems.pdf
[10] courses/math1061/graphs.pdf
[11] courses/math1061/walks-trails-and-circuits.pdf
[12] courses/math1061/graphs.pdf

- 2026-10-26-walks-trails-and-circuits[13] — Lecture 34, with worked questions on the theorem and an Euler trail on ten vertices
- 2026-07-27-welcome-and-intro-problems[14] — the bridges of Königsberg
- trees[15] — connected graphs with no non-trivial circuits
- practice-problems[16] — §44, with full worked solutions

[13] courses/math1061/2026-10-26-walks-trails-and-circuits.pdf
[14] courses/math1061/2026-07-27-welcome-and-intro-problems.pdf
[15] courses/math1061/trees.pdf
[16] courses/math1061/practice-problems.pdf