

Graph Theory

Aayush Bajaj

July 10, 2026

Contents

1 definitions	1
1.1 the handshake lemma	2
2 paths, cycles, connectivity	2
3 trees	2
4 eulerian circuits	3
5 hamiltonian cycles	4
6 planarity	4
7 colouring	5
8 matchings	5
9 the adjacency matrix	6
10 see also	7

graph theory began in 1736 when euler noticed that a walking-tour puzzle about seven bridges in königsberg was really a question about dots and lines — and that the geometry (lengths, positions, the river) was irrelevant. that act of abstraction — keep the incidence structure, discard everything else — is the whole subject. a graph is the minimal mathematical model of “things, some pairs of which are related”, which is why it keeps turning up: networks, molecules, dependencies, state machines, social graphs (Epp, Susanna S., 2019).

the first theorem of graph theory killed an entire genre of pub trivia

1 definitions

a **graph** $G = (V, E)$ is a set of **vertices** V and a set of **edges** E , each edge joining a pair of vertices. flavours:

- **simple graph**: edges are unordered pairs $\{u, v\}$, $u \neq v$, no repeats — at most one edge per pair, no loops.
- **multigraph**: parallel edges (and possibly loops) allowed. königsberg needs this — two bridges join the same pair of land masses.
- **directed graph** (digraph): edges are ordered pairs (u, v) — one-way streets, function calls, “follows” relations.

- **weighted graph**: edges carry numbers; the natural habitat of shortest-path and spanning-tree algorithms.

the **degree** $\deg(v)$ is the number of edge-ends at v (a loop counts twice; in digraphs split into in- and out-degree). a graph is **bipartite** if V splits into two parts with all edges crossing between them — equivalently (nice exercise) iff it contains no odd cycle.

1.1 the handshake lemma

theorem. in any graph or multigraph, $\sum_{v \in V} \deg(v) = 2|E|$.

proof. count the pairs (edge, end) two ways. each edge has exactly two ends, so the count is $2|E|$. grouping the same pairs by the vertex at which the end sits gives $\sum_v \deg(v)$. ■

corollary. the number of odd-degree vertices is even — the total degree sum is even, so the odd degrees must contribute an even total, and a sum of k odd numbers is even only for even k . this parity constraint is tiny but lethal: it is exactly what dooms königsberg below.

at any party, the number of people who have shaken an odd number of hands is even. hence the name

2 paths, cycles, connectivity

- a **walk** is a sequence $v_0 e_1 v_1 e_2 \cdots e_k v_k$ with each e_i joining v_{i-1} to v_i ; its **length** is k . vertices and edges may repeat.
- a **trail** is a walk with no repeated edge; a **path** is a walk with no repeated vertex.
- a **cycle** is a closed walk with no repeated vertex except the endpoints, length at least 3 in simple graphs.
- G is **connected** if every pair of vertices is joined by a path. the maximal connected pieces are the **components**.
- a **bridge** (cut edge) is an edge whose removal increases the number of components; a **cut vertex** likewise for vertices.

useful lemma: any walk from u to v contains a path from u to v — repeatedly excise the detours between repeated vertices. so “connected by a walk” and “connected by a path” coincide, which is why walk-counting with matrices (below) still detects connectivity.

3 trees

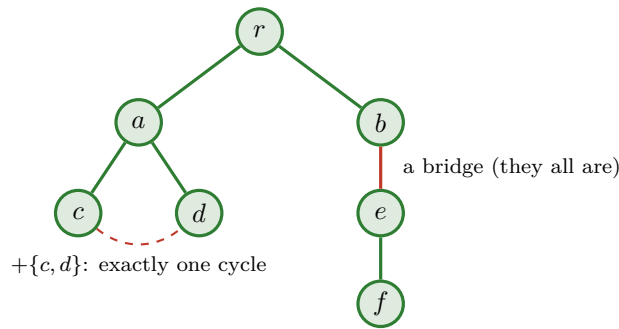
a **tree** is a connected acyclic graph. the miracle is how many different definitions land on the same object:

theorem (equivalent characterisations). for a graph T with n vertices, the following are equivalent:

1. T is connected and acyclic.
2. T is connected and has $n - 1$ edges.
3. T is acyclic and has $n - 1$ edges.
4. every pair of vertices is joined by exactly one path.

5. T is **minimally connected**: connected, but every edge is a bridge.
6. T is **maximally acyclic**: acyclic, but adding any missing edge creates exactly one cycle.

proof sketch (the load-bearing implications). $(1 \Rightarrow 2)$: induction on n — a finite acyclic connected graph has a leaf (walk forward without repeating vertices until stuck; acyclicity means “stuck” is degree 1); delete it and apply the hypothesis. $(1 \Rightarrow 4)$: a path exists by connectedness; two distinct $u-v$ paths would bound a cycle (follow one out, return along the other between the first vertex where they diverge and the next where they remeet). $(4 \Rightarrow 6)$: adding $\{u, v\}$ closes the unique $u-v$ path into a cycle, and a second cycle would yield a second path. (5) and (3) fall out the same way. ■



$n = 7$ vertices, $n - 1 = 6$ edges, unique paths everywhere

Figure 1: one tree, three characterisations at once: connected with $n - 1$ edges; adding the dashed edge creates exactly one cycle; every solid edge is a bridge.

every connected graph contains a **spanning tree** (delete edges from cycles until none remain); finding the cheapest one is the minimum-spanning-tree problem over in [classical algorithms](#).

4 eulerian circuits

back to königsberg. land masses become vertices, bridges become edges:

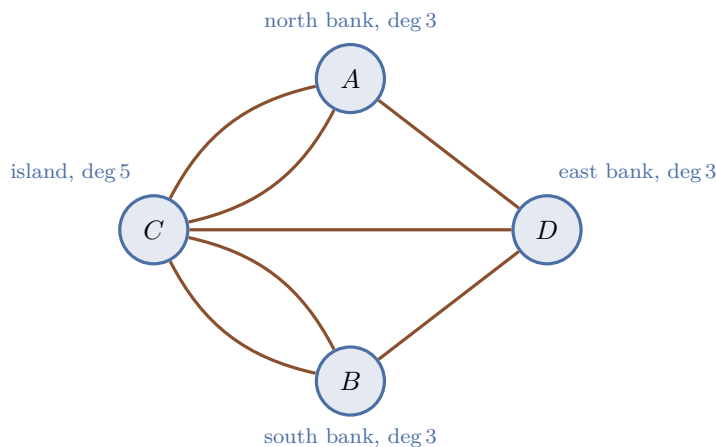


Figure 2: the seven bridges of königsberg as a multigraph. every vertex has odd degree, so no walk crosses every bridge exactly once.

an **eulerian circuit** is a closed trail using every edge exactly once; an **eulerian**

trail drops the “closed”.

euler’s theorem. a connected multigraph has an eulerian circuit iff every vertex has even degree, and an eulerian trail iff exactly zero or two vertices have odd degree.

proof sketch. **necessity:** each time a circuit passes through a vertex it consumes one edge in and one edge out, pairing up that vertex’s edge-ends — so every degree is even (for an open trail, the two endpoints escape the pairing once and may be odd). **sufficiency** (splicing): with all degrees even, walk from anywhere without reusing edges; you can only get stuck back at the start (arriving anywhere else uses an odd number of that vertex’s edge-ends, leaving one free to depart on), so you return with a closed trail. if edges remain, connectedness gives a remaining edge touching your trail at some vertex v ; degrees in the leftover graph are still all even, so grow a second closed trail from v and splice it in. repeat. ■

königsberg has degrees 5, 3, 3, 3 — four odd vertices — so neither a circuit nor a trail exists. euler’s 1736 answer to the burghers: stop trying. the splicing proof is constructive and runs in $O(|E|)$ — hierholzer’s algorithm.

euler proved necessity and asserted the rest; sufficiency was written down rigorously by hierholzer in 1873

5 hamiltonian cycles

flip the quantifier: a **hamiltonian cycle** visits every *vertex* exactly once. despite the cosmetic symmetry with euler, deciding hamiltonicity is NP-complete — no clean characterisation is known, and none is expected. what we have are sufficient conditions that force enough edges:

- **dirac (1952):** if G is simple, $n \geq 3$, and $\deg(v) \geq n/2$ for every vertex, then G is hamiltonian.
- **ore (1960):** if G is simple, $n \geq 3$, and $\deg(u) + \deg(v) \geq n$ for every *non-adjacent* pair u, v , then G is hamiltonian.

dirac is the special case of ore (two vertices of degree at least $n/2$ sum to at least n). the bounds are sharp: $K_{k,k+1}$ has minimum degree k , just below $(2k+1)/2$, and no hamiltonian cycle — a cycle must alternate sides, which forces equal part sizes.

6 planarity

a graph is **planar** if it can be drawn in the plane with no edge crossings. a planar drawing carves the plane into **faces** (regions), counting the unbounded outer one.

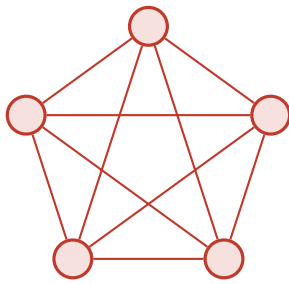
euler’s formula. for a connected planar graph drawn in the plane: $V - E + F = 2$.

proof sketch. induction on E with V fixed. base: a spanning tree has $E = V - 1$ and a single (outer) face: $V - (V - 1) + 1 = 2$. step: each further edge closes a cycle, splitting one face in two — E and F both rise by one and the alternating sum is unchanged. ■

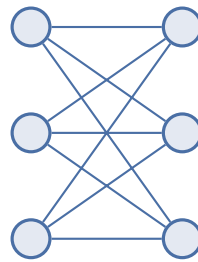
two corollaries by double-counting face–edge incidences (in a simple planar graph with $V \geq 3$, every face is bounded by at least 3 edges and every edge borders at most 2 faces, so $3F \leq 2E$):

$$E \leq 3V - 6 \quad (\text{simple planar}), \quad E \leq 2V - 4 \quad (\text{simple planar bipartite}), \quad (1)$$

the bipartite bound because odd faces are impossible there, so every face has at least 4 sides and $4F \leq 2E$. now the two famous villains:



$$K_5: V = 5, E = 10 > 3V - 6 = 9$$



$$K_{3,3}: V = 6, E = 9 > 2V - 4 = 8$$

Figure 3: the two minimal non-planar graphs. K_5 violates $E \leq 3V - 6$; $K_{3,3}$ violates the bipartite bound $E \leq 2V - 4$.

so K_5 and $K_{3,3}$ are non-planar — and remarkably, they are the *only* obstructions:

kuratowski's theorem (1930). a graph is planar iff it contains no subgraph that is a subdivision of K_5 or $K_{3,3}$. (wagner's variant: iff neither K_5 nor $K_{3,3}$ is a **minor**.) $K_{3,3}$ is the “three utilities puzzle” — three houses, three utilities, no crossing pipes — impossible, and now you know why.

a subdivision replaces edges by paths — plant extra degree-2 vertices along them

7 colouring

a **proper k -colouring** assigns each vertex one of k colours with adjacent vertices coloured differently; the **chromatic number** $\chi(G)$ is the least such k .

- **four-colour theorem:** every planar graph has $\chi \leq 4$. stated honestly: proved by appel and haken in 1976 by reducing to an unavoidable set of ~2000 configurations and checking them by computer — the first major theorem with a machine-checked case analysis at its heart, and no human-surveyable proof is known. the coq-verified proof (gonthier, 2005) at least shrinks the trusted base to a small proof kernel.
- **greedy bound:** $\chi(G) \leq \Delta(G) + 1$, where Δ is the maximum degree. proof: order the vertices arbitrarily, colour each with the smallest colour absent among its already-coloured neighbours — at most Δ colours are ever blocked. brooks' theorem sharpens this to $\chi \leq \Delta$ for connected graphs except complete graphs and odd cycles.
- **chromatic polynomial** (teaser): the number $P(G, k)$ of proper k -colourings is a polynomial in k . for a tree on n vertices $P = k(k-1)^{n-1}$ (root it, colour downward); for the triangle $P = k(k-1)(k-2)$. it satisfies deletion-contraction, $P(G) = P(G - e) + P(G/e)$, and $\chi(G)$ is the least positive integer where P is nonzero.

8 matchings

a **matching** is a set of edges with no shared endpoints; it **saturates** a vertex set if every vertex there is covered. in a bipartite graph with parts X (applicants) and Y (jobs):

hall's marriage theorem (1935). a bipartite graph has a matching saturating X

iff every subset $S \subseteq X$ satisfies $|N(S)| \geq |S|$, where $N(S)$ is the set of neighbours of S in Y .

necessity is immediate — S can only be matched into $N(S)$. sufficiency is the content: the *only* obstruction to a full matching is some set of applicants collectively interested in too few jobs. the condition naively needs $2^{|X|}$ checks, but augmenting-path algorithms find a maximum matching (or a violating set) in polynomial time; könig's theorem meanwhile equates maximum matching size with minimum vertex cover in bipartite graphs.

9 the adjacency matrix

number the vertices $1, \dots, n$ and set A_{ij} = number of edges between i and j . for simple graphs A is a symmetric 0–1 matrix; for multigraphs the entries count parallel edges.

theorem. $(A^k)_{ij}$ is the number of walks of length k from i to j .

proof. induction on k . $k = 1$ is the definition. step: a walk of length $k + 1$ from i to j is a walk of length k from i to some vertex m , followed by an edge $m-j$; summing over m gives $\sum_m (A^k)_{im} A_{mj} = (A^{k+1})_{ij}$. ■

the demo below runs this on königsberg, checks one entry by brute-force enumeration, and adds the spectral teaser: A is symmetric, so it has real eigenvalues $\lambda_1, \dots, \lambda_n$, and the total number of closed k -walks is

$$\text{tr}(A^k) = \sum_{i=1}^n \lambda_i^k \quad (2)$$

— the spectrum knows the walk counts, which is the opening move of spectral graph theory. packaging the counts as a generating function gives $\sum_{k \geq 0} A^k z^k = (I - zA)^{-1}$: walk enumeration becomes linear algebra outright, in the best concrete-mathematics tradition (Graham, Ronald L. and Knuth, Donald E. and Patashnik, Oren, 1994).

bipartite graphs have symmetric spectra; regular graphs have their degree as top eigenvalue — the spectrum is a surprisingly chatty invariant

```

1 import numpy as np
2 from itertools import product
3
4 # konigsberg as a multigraph: vertices 0=A (north bank), 1=B (south bank),
5 # 2=C (kneiphof island), 3=D (east bank). entries = number of bridges.
6 A = np.array([[0, 0, 2, 1],
7               [0, 0, 2, 1],
8               [2, 2, 0, 1],
9               [1, 1, 1, 0]])
10
11 names = "ABCD"
12 deg = A.sum(axis=1)
13 print("degrees:", {n: int(d) for n, d in zip(names, deg)}, "-> sum =",
14       ↪ int(deg.sum()),
15       "= 2 x", int(deg.sum()) // 2, "bridges (handshake lemma)")
16 print("odd-degree vertices:", [names[i] for i in range(4) if deg[i] % 2],
17       "-> no eulerian circuit or trail\n")
18
19 # walk counting: (A^k)[i,j] = number of walks of length k from i to j
20 for k in (2, 3):
21     print(f"A^{k}:\n{np.linalg.matrix_power(A, k)}")
22
23 # brute-force check of A^3[0,1]: enumerate all 3-step walks A -> B,
24 # weighting each step by the number of parallel bridges

```

```

24 count = sum(A[0, x] * A[x, y] * A[y, 1] for x, y in product(range(4),
    ↪ repeat=2))
25 print("\nbrute-force walks of length 3 from A to B:", count,
26       " (A^3)[A,B] =", np.linalg.matrix_power(A, 3)[0, 1])
27
28 # spectral teaser: closed walks of length k = sum of eigenvalues^k
29 lam = np.linalg.eigvalsh(A)
30 print("\neigenvalues:", np.round(lam, 4))
31 for k in (2, 3):
32     print(f"trace(A^{k}) = {np.trace(np.linalg.matrix_power(A, k))}")
33     f" sum(lambda^{k}) = {np.sum(lam**k):.4f}"

```

degrees: {'A': 3, 'B': 3, 'C': 5, 'D': 3} → sum = 14 = 2 × 7 bridges (handshake lemma)
 odd-degree vertices: ['A', 'B', 'C', 'D'] → no eulerian circuit or trail

A²:

```

[[5 5 1 2]
 [5 5 1 2]
 [1 1 9 4]
 [2 2 4 3]]

```

A³:

```

[[ 4  4 22 11]
 [ 4  4 22 11]
 [22 22  8 11]
 [11 11 11  8]]

```

brute-force walks of length 3 from A to B: 4 (A³)[A,B] = 4

eigenvalues: [-2.8649 -0.7685 -0. 3.6334]

trace(A²) = 22 sum(lambda²) = 22.0000

trace(A³) = 24 sum(lambda³) = 24.0000

two sanity checks hiding in the output. the diagonal of A^2 is (5, 5, 9, 3): in a loopless multigraph $(A^2)_{ii} = \sum_j A_{ij}^2$ — for vertex A that is $2^2 + 1^2 = 5$, *not* its degree 3, because a pair of parallel bridges offers 2×2 out-and-back combinations. (in a simple graph the squares collapse and the diagonal really is the degree sequence.) and the eigenvalue exactly 0: rows A and B are identical — the two banks are structurally interchangeable twins — so A is singular by construction.

the algorithmic side of all this — BFS, DFS, shortest paths, topological sort — lives in [classical algorithms](#); sedgewick (Sedgewick, Robert, 2001) is the reference implementation culture.

10 see also

- [data structures & algorithms](#) — graphs as adjacency lists, and what to do with them
- [classical algorithms](#) — BFS, DFS, dijkstra, spanning trees
- [combinatorics](#) — counting labelled trees (cayley's n^{n-2}) and colourings
- [logic](#) — the proof techniques (induction, minimal counterexample) used on every theorem above
- [sets, functions and sequences](#) — relations, of which graphs are the picture

references

Epp, Susanna S. (2019). *Discrete Mathematics with Applications*, Cengage Learning.

Graham, Ronald L. and Knuth, Donald E. and Patashnik, Oren (1994). *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley.

Sedgewick, Robert (2001). *Algorithms in C, Part 5: Graph Algorithms*, Addison-Wesley.