G-01 — Graph Definition

Def. A **graph** $G = (V, E)$ consists of:

- V: finite set of **vertices** (nodes)
- $E \subseteq V \times V$: set of **edges**

Undirected: $(u, v) = (v, u)$
Directed (digraph): $(u, v) \neq (v, u)$
→ See G-02: Degree
→ See G-05: Adjacency Matrix
#foundations #definitions

G-02 — Degree

Def. The **degree** $\deg(v)$ of vertex v is the number of edges incident to v.
Thm. (Handshaking Lemma)

$$\sum_{v \in V} \deg(v) = 2|E|$$

Corollary: Number of odd-degree vertices is even.
→ See G-01: Graph Definition
→ See G-04: Eulerian Graphs
#degree #handshaking

G-03 — Paths & Cycles

Def.

- **Walk:** sequence $v_0, e_1, v_1, \ldots, v_k$ (vertices/edges can repeat)
- **Path:** walk with no repeated vertices
- **Cycle:** closed walk with no repeated vertices except $v_0 = v_k$

Length = number of edges.
→ See G-06: Connectivity
→ See G-04: Eulerian Graphs
#paths #cycles

G-04 — Eulerian Graphs

Def.

- **Eulerian path:** visits every edge exactly once
- **Eulerian circuit:** Eulerian path that starts and ends at same vertex

Thm. (Euler, 1736)
G has Eulerian circuit $\Leftrightarrow$ G connected and all vertices have even degree.
G has Eulerian path $\Leftrightarrow$ exactly 0 or 2 vertices have odd degree.
→ See G-02: Degree (Handshaking)
#eulerian #circuits

`G-05` — Adjacency Matrix

Def. For G with n vertices, the **adjacency matrix** $A \in \{0,1\}^{n\times n}$:

$$A_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Properties:

- Symmetric for undirected graphs
- $(A^k)_{ij} = \#$ of walks of length k from v_i to v_j

→ See G-01: Graph Definition
→ See G-10: Spectral Graph Theory

`#representation #matrix`

`G-06` — Connectivity

Def.

- G is **connected** if $\exists$ path between any two vertices
- **Connected component:** maximal connected subgraph
- **Cut vertex:** removal disconnects G
- **Bridge:** edge whose removal disconnects G

Thm. G connected $\Rightarrow |E| \geq |V| - 1$

→ See G-07: Trees
→ See G-03: Paths

`#connectivity #components`

G-07 — Trees

Def. A **tree** is a connected acyclic graph.
Thm. Equivalent definitions for G with n vertices:

1. G is connected and has $n-1$ edges
2. G is acyclic and has $n-1$ edges
3. G is connected and removing any edge disconnects it
4. Any two vertices connected by unique path

→ *See G-08: Spanning Trees*
→ *See G-06: Connectivity*
#trees #acyclic

G-08 — Spanning Trees

Def. A **spanning tree** of G is a subgraph that:

- Contains all vertices of G
- Is a tree

Thm. (Cayley) Number of labeled spanning trees of K_n is n^{n-2}.
Algorithms:

- Kruskal: greedy, sort edges by weight
- Prim: grow tree from single vertex

→ *See G-07: Trees*
#spanning #mst

G-09 — Planar Graphs

Def. G is **planar** if it can be drawn in the plane without edge crossings.
Thm. (Euler's Formula) For connected planar graph:

$$\boxed{V - E + F = 2}$$

where F = number of faces (including outer).
Cor. For planar G with $|V| \geq 3$: $|E| \leq 3|V| - 6$
Non-planar: K_5, $K_{3,3}$
→ *See G-11: Kuratowski*
#planar #euler

G-10 — Graph Coloring

Def. A **proper k-coloring** assigns colors $\{1, \ldots, k\}$ to vertices s.t. adjacent vertices have different colors.
Def. **Chromatic number** $\chi(G)$ = minimum k for proper coloring.
Bounds:

- $\chi(G) \geq \omega(G)$ (clique number)
- $\chi(G) \leq \Delta(G) + 1$ (max degree + 1)

Thm. (4-Color) Every planar graph is 4-colorable.
→ *See G-09: Planar Graphs*
#coloring #chromatic

G-11 — Bipartite Graphs

Def. $G = (V, E)$ is **bipartite** if $V = A \cup B$ with $A \cap B = \emptyset$ and all edges go between A and B.
Thm. G is bipartite $\Leftrightarrow$ G has no odd cycles $\Leftrightarrow$ $\chi(G) \leq 2$.
Examples:

- Trees are bipartite
- $K_{m,n}$ = complete bipartite

→ *See G-10: Graph Coloring*
→ *See G-12: Matching*
#bipartite #2colorable

G-12 — Matching

Def. A **matching** $M \subseteq E$ is a set of edges with no shared vertices.
Perfect matching: covers all vertices.
Thm. (Hall's Marriage) Bipartite $G = (A \cup B, E)$ has matching covering A $\Leftrightarrow$

$$|N(S)| \geq |S| \quad \forall S \subseteq A$$

where $N(S)$ = neighbors of S.
Algorithm: Hungarian (poly-time for bipartite).
→ *See G-11: Bipartite*
#matching #hall