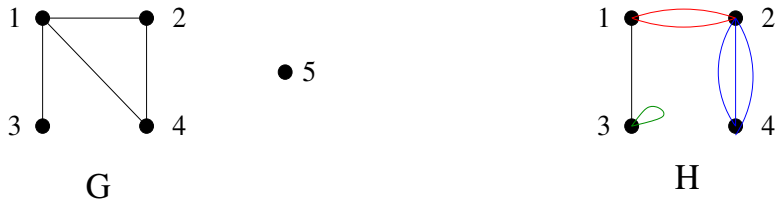
1. The Basics

1.1. Graphs.

Definition 1.1. A **graph** G is a pair (V, E), where V is a (finite), nonempty set of **vertices** and E is a (finite) set of **edges**. Each edge e is given by an unordered pair of two (possibly equal) vertices v, w, called its **endpoints**.



Equivalent statements:

- v, w are the **endpoints** of e; or
- v, w are **joined** by the edge e; or
- $e = vw$.

Technically, this last notation should only be used when e is the *only* edge joining v and w. Note that $e = wv$ is equivalent.

Sometimes, we don't want to bother to give the edge e a name; it is enough to know that there exists *some* edge joining v and w. Then we might say that v, w are **adjacent** or are **neighbors**. (It's tempting to say "connected" instead, but you should try to make a habit of resisting temptation, because that term properly means something else.)

Graphs can have **loops** (edges that join a vertex to itself) and **parallel edges** (edges with the same pairs of endpoints). Sometimes we want to exclude these possibilities, often because they are irrelevant. A graph with no loops and no parallel edges is called **simple**.

When studying graph theory, one quickly learns to be flexible about notation. For instance, when working with a single graph we want to use the concise symbols V and E for its vertex and edge sets; but if there are several different graphs around then it is clearer to write $V(G)$, $E(H)$, etc.

1.2. Isomorphisms and subgraphs.

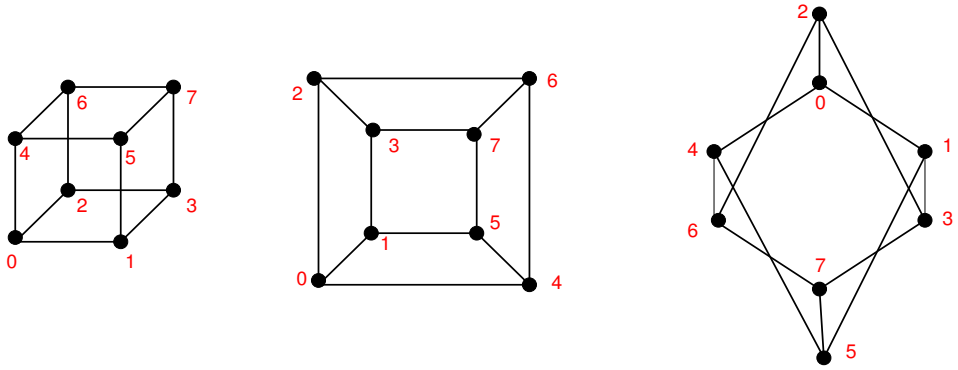
As in man fields of mathematics, one of our first orders of business is to say when two of the things we want to study are the same, and when one is a subthing of another thing.

Definition 1.2. Let G, H be graphs. An **isomorphism** is a bijection $f : V(G) \to V(H)$ such that for every $v, w \in V(G)$,

$$\#\{\text{edges of } G \text{ joining } v, w\} \;=\; \#\{\text{edges of } H \text{ joining } f(v), f(w)\}.$$

"$G \cong H$" means G, H are isomorphic.

Notice that this has nothing to do with what the graph looks like on the paper. **A drawing of a graph is not the same as the graph itself!** These three graphs are all isomorphic to each other; the red numbers indicate the isomorphism.



Think of an isomorphism as a relabeling, which doesn't really change the underlying structure of the graph.

Definition 1.3. An **isomorphism invariant** is a function ψ on graphs such that $\psi(G) = \psi(H)$ whenever $G \cong H$. (Equivalently, a function on equivalence classes of graphs.)

For example, the number of vertices is an invariant, as is the number of edges — but not the number of crossings when you draw the graph (although the minimum number of crossings among all possible drawings is indeed an invariant). Nor is a property like "Every edge has one vertex labeled with an odd number and one vertex labeled with an even number," since there's nothing to prevent me from shuffling the numbers to make this false. On the other hand, "The graph *can* be labeled so that every edge has one odd and one even label" is an invariant.

It is always possible to draw a given graph in lots of different ways, many geometrically inequivalent. It is crucial to remember that *a graph is a combinatorial object, not a geometric one.* That is, the structure of a graph really is given by its list of vertices, edges and incidences, *not* by a drawing composed of points and lines.

Definition 1.4. Let G be a graph. A **subgraph** of G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For short we write $H \subseteq G$.

Note that it is *not* true that for every $X \subseteq V(G)$ and $F \subseteq E(G)$, the pair (X, F) is a subgraph of G, because it might not even be a graph—it needs to satisfy the condition that every endpoint of an edge in F belongs to X. (You can't have an edge dangling in the breeze — there needs to be a vertex on each end of it.)

Every subset $F \subseteq E(G)$ determines a subgraph whose vertex set is the set of all vertices that are endpoints of at least one edge in F. Also, every subset $X \subseteq V(G)$ determines a subgraph $G[X]$, the **induced subgraph**, whose edge set is the set of *all* edges of G with both endpoints in X. Being an induced subgraph is a stronger property than being a subgraph.

1.3. **Some applications of graph theory.** Graph theory has about a zillion applications. Here are a few.

Discrete optimization: a lot of discrete optimization problems can be modeled using graphs. For example, the TSP (traveling salesperson problem); the knapsack problem; matchings; cuts and flows.

Discrete geometry and linear optimization: the vertices and edges of a polytope P form a graph called its *1-skeleton*; when using the simplex method to solve a linear programming problem whose feasible region (i.e., the set of legal, although perhaps not optimal, solutions) is P, the 1-skeleton of P describes exactly the steps of the algorithm.

Algebra: the *Cayley graph* of a group G is a graph whose edges correspond to multiplication by one of a given set of generators; basic group-theoretic notions such as relations, conjugation, etc. now have natural descriptions in terms of the Cayley graph.

Topology: you can study an infinite and therefore complicated topological space by replacing it with a finite simplicial complex (a generalized kind of graph) from which you can calculate properties of the original space; also, deep graph-theoretic concepts such as deletion/contraction often have topological analogues.

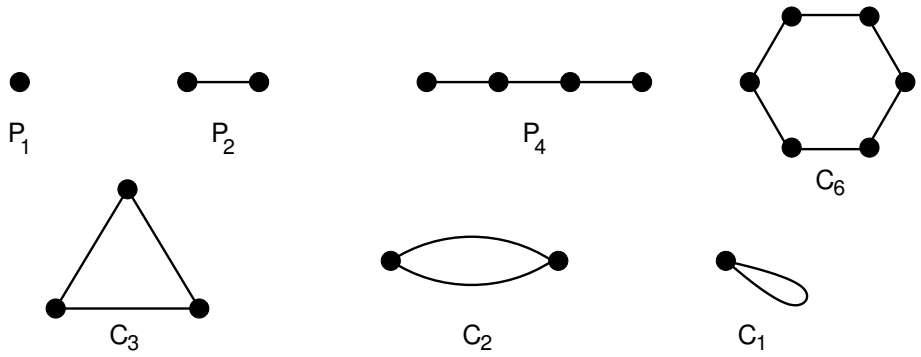
Theoretical computer science: many fundamental constructions such as finite automata are essentially glorified graphs, as are data structures such as binary search trees.

Chemistry: A molecule can be regarded as a graph in which vertices are atoms and edges are bonds. Amazingly, the chemical properties of a substance, such as its boiling point, can sometimes be predicted with great accuracy from the purely mathematical properties of the graph of the molecule!

Biology: More complicated structures like proteins can be modeled as graphs. The theory of rigidity of graphs has been used to understand how proteins fold and unfold.

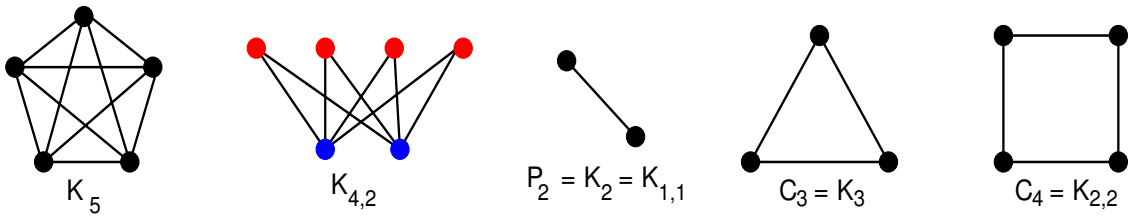
Not to mention the wonderful applicability of graphs to all manner of subjects including forestry, communications networks, efficient garbage collection, and evolutionary biology.

1.4. **Some important graphs and basic constructions.** The **path** P_n (Diestel: P^n) has n vertices and $n-1$ edges, connected sequentially. The **cycle** C_n (Diestel: C^n)has n vertices and n edges and can be drawn as a polygon.



The **complete graph** K_n (Diestel: K^n) has n vertices and one edge between each pair of vertices. Thus there are $\binom{n}{2} = \frac{n(n-1)}{2}$ edges in total. Often we assume that the vertex set is $[n] = \{1, 2, \ldots, n\}$. (This notation is standard in combinatorics.) A complete graph is also called a **clique**, particularly when it occurs inside another graph.

The **complete bipartite graph** $K_{p,q}$ has $p + q$ vertices, with p of them painted red and q painted blue, and an edge between each pair of differently colored vertices, for a total of pq edges.



The **empty graph** or $\bar{K}_n$ consists of n vertices and no edges. A copy of $\bar{K}_n$ appearing as an induced subgraph of a graph G is the same as a set of vertices of G of which no two are adjacent. Such a set is called a **coclique** (or **independent set** or **stable set**).

A few operations on graphs.

- If G is a simple graph, its **complement** $\bar{G}$ is the graph obtained by toggling adjacency and non-adjacency.
- The **underlying simple graph** G^{s} of any graph G is obtained by deleting all loops and all but one element of each parallel class of edges. Note that the connectivity relation on G^{s} is the same as that on G.
- The **disjoint union** $G + H$ is the union of G and H, assuming that the vertex sets are disjoint. For example, $\bar{K}_n + \bar{K}_m = \overline{K_{m+n}}$ and $\overline{K_{m,n}} = K_m + K_n$.
- The **join** $G * H$ also has vertex set $V(G) \cup V(H)$, but this time we add every possible edge between a vertex of G and a vertex of H.