INTRODUCTORY GRAPH THEORY

CONTENTS

SECTION I. INTRODUCTION

I.A. ENVIRONMENTS FOR GRAPHS: SETS AND MULTISETS

I.A.1. SETS

STANDARD OPERATIONS

We follow the ZFC model of set theory (built from the Zermelo-Fraenkel axioms together with the axiom of choice) when discussing sets.

In ZFC, we assume the presence of a set U called the **universe of discourse;** in practice, U is constructed from familiar sets such as $\mathbb{Z}$ or $\mathbb{R}$. Every element x is assumed to be an element of U.

We recall that two sets A and B are **equal** when for all $x \in U$, $x \in A$ iff $x \in B$. We will write $A = B$ for this relationship.

We recall that a set A is a **subset** of B when for all $x \in U$, if $x \in A$, then $x \in B$. We will write $A \subseteq B$ for this relationship. Notice that $A = B$ is the same as $A \subseteq B$ and $B \subseteq A$.

The usual set operations of union, intersection, and complement are defined as follows, given a universe of discourse U that contains all possible elements under consideration:

- The **union** of two sets A and B is the set $A \cup B = \{x \in U: x \in A \text{ or } x \in B\}$.
- The **intersection** of two sets A and B is the set $A \cap B = \{x \in U:\ x \in A \text{ and } x \in B\}$.
- The **complement** of A is the set $\overline{A} = \{x \in U:\ x \notin A\}$.

An important set is the **empty set** $\emptyset = \overline{U}$.

Other common set operations are defined in terms of the union, intersection, and complement:

- Given sets A and B, their **difference** $A - B$ is the set $A \cap \overline{B} = \{x \in U: x \in A \text{ and } x \notin B\}$.
- The **symmetric difference** of two sets A and B is the set $(A - B) \cup (B - A)$; this set is denoted $A \,\Delta\, B$.

The symmetric difference has many equivalent interpretations:

- $A \,\Delta\, B = \{x \in U:\ x \in (A \cup B) - (A \cap B)\}$
- $A \,\Delta\, B = \{x \in U: x \text{ is an element of exactly one of } A \text{ or } B\}$

Another common operation is the **Cartesian product**: Given sets A and B, their Cartesian product is the set

$$A \times B = \{(a, b):\ a \in A \text{ and } b \in B\}.$$

This can be extended to several sets:

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n):\ a_i \in A_i \text{ for all } i, 1 \leq i \leq n\}.$$

An element $(a_1, a_2, \ldots, a_n)$ of $A_1 \times A_2 \times \cdots \times A_n$ is called an **ordered *n*-tuple.**

UNIONS AND INTERSECTIONS OVER FAMILIES OF SETS; INDEXED SETS

A **family** of sets is a collection $\mathcal{A}$ of subsets of some common universe U; it is allowed for the same set to appear more than once in $\mathcal{A}$.

We define the **union of the family $\mathcal{A}$**

$$\bigcup_{A \in \mathcal{A}} A = \{x \in U: \text{ there exists } A \in \mathcal{A} \text{ such that } x \in A\}$$

and the **intersection of the family $\mathcal{A}$**

$$\bigcap_{A \in \mathcal{A}} A = \{x \in U: \ x \in A \text{ for every } A \in \mathcal{A}\}.$$

We define a family of **indexed sets** $\mathcal{A}$ where A_i is in $\mathcal{A}$ for each $i \in I$, called the **index set**. Here,

$$\bigcup_{i \in I} A_i = \{x \in U: \text{ there exists } i \in I \text{ such that } x \in A_i\}$$

and

$$\bigcap_{i \in I} A_i = \{x \in U: \ \ x \in A_i \text{ for every } i \in I\}.$$

CARDINALITY

Formally, two sets A and B are said to have the **same cardinality** if there exists a bijection $f: A \to B$; recall that a bijection is a function that is both one-to-one and onto. Another name for "bijection" is **perfect matching**. We denote the cardinality of a set A by $|A|$.

If there is a one-to-one function $f: A \to B$, then $|A| \leq |B|$.

If there is an onto function $f: A \to B$, then $|A| \geq |B|$.

A set A is **countably infinite** if $|A| = |\mathbb{Z}^+|$.

A set is **countable** if it is finite or countably infinite. A set is **uncountable** otherwise.

Observations:

- $|\emptyset| = 0$

- $|\{1,2,3,\dots,n\}| = n$
- The cardinality of a finite set is the number of its elements.
- If A is countably infinite, then one can consider A as a sequence of its elements: $A = \{a_1, a_2, a_3, \dots\}$.
- For finite sets, $|A \times B| = |A| \cdot |B|$.
- Any subset of a countable set is countable.
- Any union of countably many countable sets is countable.
- If A is uncountable and B is countable, then $A - B$ is uncountable.
- $\mathbb{Q}$ is countable
- $\mathbb{R}$ is uncountable

CHARACTERISTIC FUNCTIONS FOR SETS

Characteristic functions serve as useful algebraic tools for analyzing element/set behavior. Given a universe of discourse U and a set A, we define the **characteristic function** $\chi_A: U \to \mathbb{Z}$ by the formula

$$\chi_A(e) = \begin{cases} 1, & e \in A \\ 0, & e \notin A \end{cases}$$

We observe that

$$\chi_{\overline{A}} = 1 - \chi_A$$
$$\chi_{A \cap B} = \chi_A \, \chi_B$$
$$\chi_{A \cup B} = \chi_A + \chi_B - \chi_A \chi_B$$

The following result allows us to develop Boolean algebra in terms of real-valued algebra.

Theorem 1. For any sets A and B, we have $A = B$ iff $\chi_A = \chi_B$.

Proof. The following are equivalent:

1) $A = B$
2) For all $e \in U$, $e \in A$ iff $e \in B$
3) For all $e \in U$, $\chi_A(e) = 1$ iff $\chi_B(e) = 1$ and $\chi_A(e) = 0$ iff $\chi_B(e) = 0$
4) For all $e \in U$, $\chi_A(e) = \chi_B(e)$
5) $\chi_A = \chi_B$

Corollary 2. (Double complement) For any set A, $\overline{\overline{A}} = A$.

Proof. We have

$$\chi_{\overline{\overline{A}}} = 1 - \chi_{\overline{A}} = 1 - (1 - \chi_A) = \chi_A.$$

Observation 3. For any set A, $\chi_A^2 = \chi_A$.

Corollary 4. (Idempotency of union and intersection) For any set A, $A \cap A = A$ and $A \cup A = A$.

Proof. We have

$$\chi_{A\cap A} = \chi_A^2 = \chi_A,$$
$$\chi_{A\cup A} = \chi_A + \chi_A - \chi_A^2 = \chi_A.$$

Corollary 5. (DeMorgan's laws) For any sets A and B,

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$
$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

Proof. We have

$$\begin{aligned}\chi_{\overline{A}\cup\overline{B}} &= \chi_{\overline{A}} + \chi_{\overline{B}} - \chi_{\overline{A}}\chi_{\overline{B}}\\ &= (1-\chi_A) + (1-\chi_B) - (1-\chi_A)(1-\chi_B)\\ &= 1 - \chi_A\chi_B\\ &= 1 - \chi_{A\cap B}\\ &= \chi_{\overline{A\cap B}}\end{aligned}$$

and

$$\begin{aligned}\chi_{\overline{A}\cap\overline{B}} &= \chi_{\overline{A}}\chi_{\overline{B}}\\ &= (1-\chi_A)(1-\chi_B)\\ &= 1 - (\chi_A + \chi_B - \chi_A\chi_B)\\ &= 1 - \chi_{A\cup B}\\ &= \chi_{\overline{A\cup B}}\end{aligned}$$

The equations $\overline{A \cap B} = \overline{A} \cup \overline{B}$ and $\overline{A \cup B} = \overline{A} \cap \overline{B}$ follow from theorem 1.

Observation 6. For a finite set A,

$$\begin{aligned}\sum_{e\in U}\chi_A(e) &= \sum_{e\in A}\chi_A(e) + \sum_{e\in\overline{A}}\chi_A(e)\\ &= \sum_{e\in A}1 + \sum_{e\in\overline{A}}0\\ &= |A|.\end{aligned}$$

THE INCLUSION-EXCLUSION PRINCIPLE

DeMorgan's Laws apply to larger unions and intersections:

$$\overline{\left(\bigcup_{i\in I}A_i\right)} = \bigcap_{i\in I}\overline{A_i}$$
$$\overline{\left(\bigcap_{i\in I}A_i\right)} = \bigcup_{i\in I}\overline{A_i}$$

In particular, if $I = \{1,2,3,\dots,n\}$, then the first of these statements can be written

$$\overline{A_1 \cup A_2 \cup \cdots \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n}.$$

We now have

$$1 - \chi_{A_1 \cup A_2 \cup \cdots \cup A_n} = \prod_{i=1}^{n} (1 - \chi_{A_i})$$

or with some rearranging,

$$\begin{aligned}
\chi_{A_1 \cup A_2 \cup \cdots \cup A_n} &= 1 - \prod_{i=1}^{n} (1 - \chi_{A_i}) \\
&= 1 - \left(1 - \sum_{i=1}^{n} \chi_{A_i} + \sum_{1 \le i < j \le n} \chi_{A_i} \chi_{A_j} - \sum_{1 \le i < j < k \le n} \chi_{A_i} \chi_{A_j} \chi_{A_k} + \cdots \right) \\
&= \sum_{i=1}^{n} \chi_{A_i} - \sum_{1 \le i < j \le n} \chi_{A_i \cap A_j} + \sum_{1 \le i < j < k \le n} \chi_{A_i \cap A_j \cap A_k} - \cdots
\end{aligned}$$

Using Observation 6 from the previous section,

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{i=1}^{n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots$$

which is the **inclusion-exclusion principle.**

I.A.2. MULTISETS

As sets, $\{a, a, a\}$ and $\{a\}$ are equal. Listing an element more than once does not change the set. In contrast, a **multiset** takes the multiplicity of elements into account. The multisets $\{a, a, a\}$ and $\{a\}$ are distinct; in the first, the element a has multiplicity 3 while in the second, the element a has multiplicity 1.

Given a multiset A, we define the **multiplicity function** $\mu_A: U \to \mathbb{Z}^{\geq 0}$ where for $e \in U$, $\mu_A(e)$ is the multiplicity of e in A. This is the analog for multisets to the characteristic function χ_A for sets.

The operations of union and intersection extend naturally to multisets:

If A and B are multisets, then.

- $A \cup B$ is the multiset where $\mu_{A \cup B} = \max(\mu_A, \mu_B)$,
- $A \cap B$ is the multiset where $\mu_{A \cap B} = \min(\mu_A, \mu_B)$.

We say $A \subseteq B$ if for all $e \in U$, $\mu_A(e) \leq \mu_B(e)$.

Additionally, we can define the **sum** of multisets:

- $A + B$ is the multiset where $\mu_{A+B} = \mu_A + \mu_B$.

The complement of a multiset is not defined.

I.B. MAIN INGREDIENTS FOR A GRAPH: VERTICES AND EDGES

I.B.1. VERTICES AND EDGES

At its core, a graph describes connections between objects. The objects are called **vertices** (represented by the disks in Figure I.B.1.a) and the connections between them are called **edges** (represented by the segments or curves in Figure I.B.1.a.)

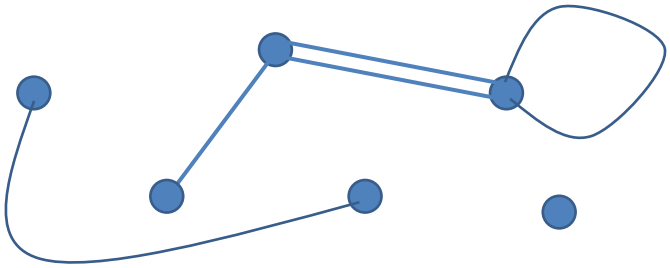


Figure I.B.1.a. An example of a graph

Figure I.B.1.b suggests an immediate connection between graph theory and organic chemistry.

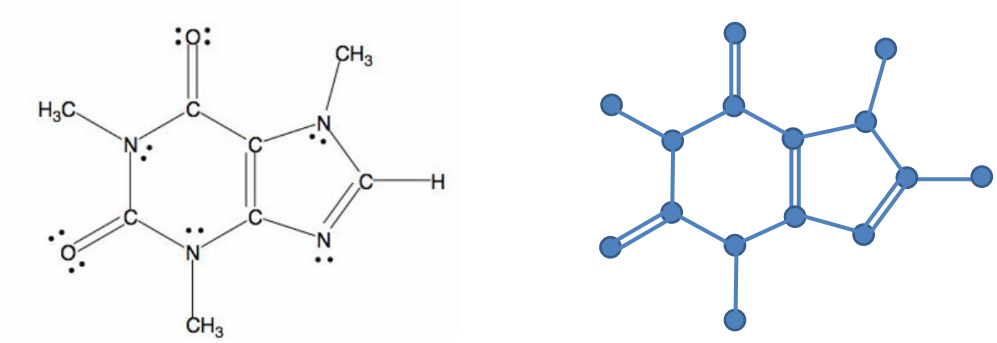


Figure I.B.1.b. Caffeine and an Associated Graph

The caffeine molecule depicted in Figure I.B.1.b can be thought of as a graph where each atom or CH_3 group is a vertex (the small dots representing electron pairs don't count) and each bond depicted as a segment (the double bonds being pairs of parallel edges) is an edge.

Definition. We define a **graph** G to be a pair (V, E) where V is a set and E is a multiset of unordered pairs of elements of V.

Definition. Two graphs G and H are **equal,** denoted $G = H$, if and only if they have the same vertex set and the same edge multiset.

The collection of objects whose connections are described by a graph is the set V of **vertices.** Note: The singular form of "vertices" is **vertex**. The set V can be *any* set. A common notational practice is to denote vertices using letters such as u, v, w, x, y, z. We'll use subscripts quite commonly, especially when the number of vertices is large or not specified. For instance, we will commonly think of V as a list:

$$V = \{v_1, v_2, v_3, \dots, v_n\}.$$

The quantity $|V|$ is called the **order** of the graph G. It is customary to reserve the variable n for the order of a graph.

The connections between vertices are encoded by a multiset E of **edges**; we allow for finitely many repeated instances of edges in E. Edges are very commonly denoted using the letter e, using subscripts when needed.

The quantity $|E|$ is the **size** of G. It is customary to reserve the variable m for the size of a graph.

Formally, an edge is an unordered pair of (not necessarily distinct) vertices. We observe that E is a multiset of sets and we allow E to be empty. Suppose $e = \{u, v\}$ is an edge. To facilitate readability, we'll shorten this to $e = uv$.

Definition. Each of the following statements conveys the same information about an edge $e \in E$:

- $e = \{u, v\}$
- $e = uv$
- $e = vu$
- u and v are the **endpoints** of e
- e is **incident** with u and v (and no other vertices are incident with e)
- u and v are **incident** with e
- e **joins** u and v

If e_1 and e_2 have the same endpoints, but are different instances in the multiset E, then they are said to be **parallel edges**.

If e_1 and e_2 share an endpoint, then they are said to be **adjacent edges**.

Each of the following statements conveys the same information about two vertices u and v:

- $uv \in E$
- u is **adjacent** to v
- v is adjacent to u
- u and v are adjacent
- u and v are **neighbors**

If u is adjacent to itself, then the edge $e = uu$ is called a **loop**.

For any vertex v, its **neighborhood** is $N(v) = \{u \in V: uv \in E\}$. If we wish to include v, we'll use

$$N_0(v) = N(v) \cup \{v\}.$$

The set $N_0(v)$ is called the **closed neighborhood** of v.

Graphs can have various properties:

- A **loopless** graph has no loops
- A **null** graph has no vertices, i.e., $V = \emptyset$. Not all authors allow this to be a graph; many theorems in graph theory flow more naturally by disallowing the null graph.
- A **simple** graph is not a null graph, has no loops, and has no parallel edges
- An **empty** graph has no edges, i.e., $E = \emptyset$
- A **trivial** graph is simple and has exactly one vertex; any other graph is said to be **nontrivial**
- An empty and nontrivial graph is said to be **totally disconnected**
- A **finite** graph is one where V is finite (it's very rare to discuss graphs where V is finite and E is infinite)
- An **infinite** graph is one where V is infinite

I.C. CONSTRUCTING AND DRAWING GRAPHS

I.C.1. CONVENTIONS

The usual conventions for drawing graphs include

- Using small circles or other shapes to depict vertices,
- Using segments or curves to depict edges,
- An edge $e = uv$ is drawn having u and v as endpoints and it does not pass through any other vertex; edges may be drawn crossing other edges, however.

Graph drawing will be discussed more formally in Section IV.

I.C.2. CONSTRUCTING A GRAPH; COMMON FAMILIES OF GRAPHS

A customary way to construct a graph without parallel edges is to specify its vertex set V and then to declare a rule that declares precisely when two vertices are adjacent. Some common examples:

Example a. For a positive integer n, the **path graph** P_n (often just called a path, but this term is reserved to the definition in II.A.2) has vertex set $V = \{1,2,3, \dots, n\}$ where two vertices in V are adjacent if they differ by exactly 1.

Figure I.C.2.a shows P_3 and P_4. **Note:** Some authors use the subscript to denote the number of edges in a path.

Figure I.C.2.a. The path graphs P_3 and P_4

Example b. For a positive integer $n \geq 3$, the **cycle graph** C_n (also called an n-**cycle**; again, this is reserved to the definition in II.A....) has vertex set $\mathbb{Z}_n = \{0,1,2, \dots, n-1\}$. Here, the arithmetic on $\mathbb{Z}_n$ is done modulo n. Two vertices in C_n are adjacent iff they differ by exactly 1. The distinction between C_n and P_n is that vertices 0 and $n-1$ are adjacent; these values differ by exactly 1 in modulo n arithmetic.

Figure I.C.2.b shows C_3, often called a **triangle**, and C_4.

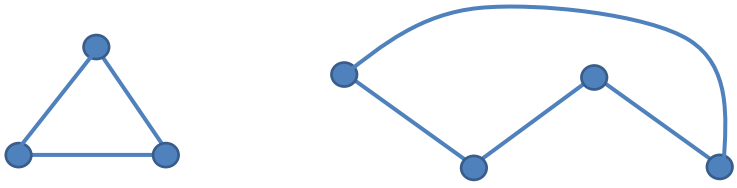


Figure I.C.2.b. The cycle graphs C_3 and C_4

It is common to define the 1-cycle graph C_1 as a single vertex with a single loop and the 2-cycle graph C_2 as two vertices joined by two parallel edges.

Example c. For a positive integer n, the **complete graph** K_n has vertex set $V = \{1,2,3,\dots,n\}$ where two vertices are adjacent iff they are distinct. The complete graphs K_1, K_2, K_3, K_4 are depicted in Figure I.C.2.c.

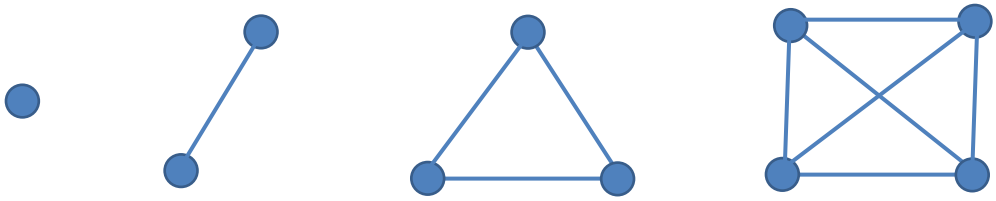


Figure I.C.2.c. The complete graphs K_1, K_2, K_3 and K_4

More generally, a graph G is **complete** if every two distinct vertices of G are adjacent.

Example d. For positive integers p, q, the **complete bipartite graph** $K_{p,q}$ has vertex set $V = \{w_1, w_2, \dots, w_p, x_1, x_2, \dots, x_q\}$; the sets $W = \{w_1, w_2, \dots, w_p\}, X = \{x_1, x_2, \dots, x_q\}$ are called **parts**. Two vertices are adjacent if and only if they belong to distinct parts. Figure I.C.2.d shows $K_{3,4}$.

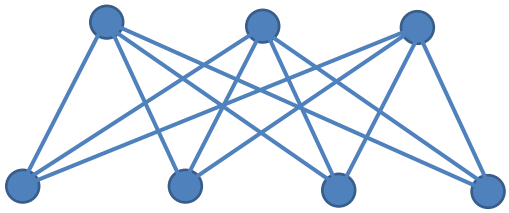


Figure I.C.2.d. The complete bipartite graph $K_{3,4}$

More generally, a graph is said to be **bipartite** if there exist sets W and X where $W \cup X = V$, $W \cap X = \emptyset$ and every edge has an endpoint in W and an endpoint in X.

Example e. For a positive integer n, the n-**cube** graph Q_n has vertex set $V = \{(a_1, a_2, \dots, a_n): a_i \in \{0,1\}\}$. Two vertices $(a_1, a_2, \dots, a_n)$ and $(b_1, b_2, \dots, b_n)$ are adjacent if and only if they differ in exactly one coordinate. Figure I.C.2.e shows Q_2 and Q_3.

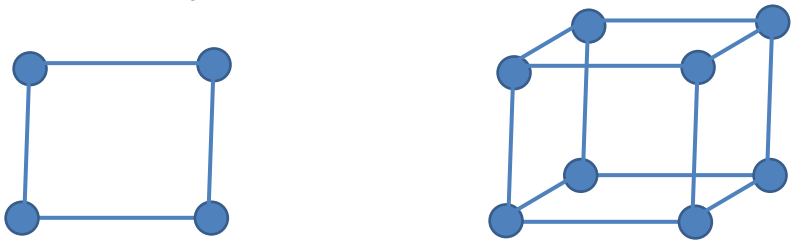


Figure I.C.2.e. The graphs Q_2 and Q_3

It turns out that for all $n \geq 1$, Q_n is bipartite.

Theorem. For all positive integers n, the graph Q_n is bipartite.

Proof. We partition the vertex set $V = \{(a_1, a_2, \ldots, a_n)\colon\ a_i \in \{0,1\}\}$ into

$$W = \left\{(a_1, a_2, \ldots, a_n) \in V\colon\ \sum_{i=1}^{n} a_i \text{ is even}\right\} \text{ and } X = \left\{(a_1, a_2, \ldots, a_n) \in V\colon\ \sum_{i=1}^{n} a_i \text{ is odd}\right\}.$$

For any pair of adjacent vertices $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$, there is a unique $k, 1 \leq k \leq n$ such that $|a_k - b_k| = 1$. Without loss of generality, we suppose $a_k - b_k = 1$. For all other $i, 1 \leq i \leq n$, $a_i - b_i = 0$. We then have

$$1 = \sum_{i=1}^{n} (a_i - b_i) = \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} b_i$$

and so $\sum_{i=1}^{n} a_i$ and $\sum_{i=1}^{n} b_i$ have opposite parities. Hence, one of $(a_1, a_2, \ldots, a_n)$ or $(b_1, b_2, \ldots, b_n)$ is an element of W and the other is an element of X and so Q_n is bipartite.

□

I.D. DEGREE

I.D.1. DEFINITIONS AND THE HANDSHAKING LEMMA

Let $G = (V, E)$ be a finite graph.

Definition. For $v \in V$, the **degree of v**, denoted $\deg(v)$, is the number of edges incident with v.

Notation. If $V = \{v_1, v_2, \dots, v_n\}$, then $d_i = \deg(v_i)$.

Observation. If G is a simple graph, then

$$\deg(v) = |N(v)|,$$

the number of neighbors of v.

A vertex v is **isolated** if $\deg(v) = 0$.

A vertex v of a loopless graph is a **leaf** or an **end vertex** if $\deg(v) = 1$. The edge it is incident with is known as a **pendant edge.**

Observation. If G is a loopless graph, then for $v \in V$,

$$\deg(v) = \sum_{u \in V} \mu_E(uv),$$

recalling that $\mu_E(uv)$ is the number of occurrences of uv in the edge multiset E.

Also, if $V = \{v_1, v_2, \dots, v_n\}$,

$$|E| = \sum_{1 \le i < j \le n} \mu_E(v_i v_j) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mu_E(v_i v_j).$$

As a consequence of this observation, we have "the first theorem of graph theory," often stated, "the total degree is twice the number of edges."

Theorem I.D.1.a. If G is a loopless graph, then

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

Corollary I.D.1.b. The total degree of a finite simple graph is even.

Corollary I.D.1.c. In any loopless graph, there are an even number of vertices with odd degree.

This theorem is also called the "handshaking lemma": Suppose everyone attending a party (very carefully) counts the number of hands they shake during the evening. If the individual counts are totaled, the sum is twice the total number of handshakes.

I.D.2. BOUNDS ON DEGREES AND REGULARITY

Definition. Given a loopless graph G, the **minimum degree** $\delta(G)$ is given by $\delta(G) = \min\{\deg(v): v \in V\}$ and the **maximum degree** $\Delta(G)$ is given by $\Delta(G) = \max\{\deg(v): v \in V\}$. If $\{\deg(v): v \in V\}$ has no upper bound, then $\Delta(G) = \infty$.

Definition. The graph G is said to be r-**regular** if any of the following equivalent conditions hold:

- for all $v \in V, \deg(v) = r$
- $\delta(G) = \Delta(G) = r$
- $\{\deg(v) : v \in V\} = \{r\}$

Observations.

- A **totally disconnected** graph is 0-regular. All vertices of a totally disconnected graph are isolated vertices.
- A 1-regular graph consists of a disjoint collection of edges. To continue with the chemistry analogy, a 1-regular graph resembles a sample of hydrogen gas, consisting entirely of H_2 molecules.
- All cycle graphs C_n for $n \geq 2$ are 2-regular.
- A finite 2-regular graph consists of a disjoint collection of cycles. An infinite 2-regular graph may include copies of a path $P_{\mathbb{Z}}$ that stretches infinitely in both directions.
- A 3-regular graph is often called a **cubic** graph. This name arises from the fact that the 3-cube Q_3 is 3-regular. Cubic graphs have been studied extensively as this class of graphs spans a wide variety of structural properties. In a sense, they are at the boundary of where graph theory becomes "interesting," from a theoretical standpoint.
- The complete graph K_n is $(n-1)$-regular; every vertex is adjacent to $n-1$ other vertices.
- The cube graph Q_n is n-regular
- The complete bipartite graph $K_{n,n}$ is n-regular. If $a \neq b$, then $K_{a,b}$ is not regular.
- If G is a loopless d-regular graph with n vertices and m edges, then $2m = nd$. If G has loops where each loop contributes 2 to the degree of its vertex, then this formula still holds.

I.D.3. DEGREE SEQUENCES AND DEGREE-COUNT SEQUENCES

Definition. Suppose we have a finite loopless graph G whose vertex set $V = \{v_1, v_2, \dots, v_n\}$ is indexed so that if $i < j$, then $d_i \geq d_j$, i.e., the vertices are listed in descending order of their degrees. The sequence of values

$$d_1, d_2, \dots, d_n$$

is known as the **degree sequence** of G.

Theorem (Erdos and Gallai, 1960). A sequence $d_1 \geq d_2 \geq \cdots \geq d_n \geq 1$ is a degree sequence for some graph without isolated vertices if and only if $\sum_{i=1}^{n} d_i$ is even and for all $r, 1 \leq r \leq n-1$,

$$\sum_{i=1}^{r} d_i \leq r(r-1) + \sum_{i=r+1}^{n} \min(r, d_i).$$

Definition. Given a loopless graph G, we define n_i as the number of vertices of G with degree i. The infinite sequence of values

$$n_0, n_1, n_2, \ldots$$

is called the **degree-count sequence.**

Observations.

$$\sum_{i=0}^{\infty} n_i = n; \quad \sum_{i=0}^{\infty} i n_i = \sum_{j=1}^{n} d_j = \sum_{v \in V} \deg(v) = 2m.$$

Definition. For a non-null finite graph, the **average degree** is the quantity $d_{\text{ave}} = \frac{1}{n}\sum_{j=1}^{n} d_j$. We have

$$\delta \leq d_{\text{ave}} \leq \Delta.$$

Observation. If $0 < \delta$ (equivalently, G has no isolated vertices) and $d_{\text{ave}} < 2$, then $\delta = 1$.

I.E. SUBGRAPHS

I.E.1. GENERAL DEFINITION

Given graphs G, H with vertex sets V_G, V_H and edge multisets E_G, E_H, respectively, we say H is a **subgraph** of G if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. If H is a subgraph of G, then we write $H \leq G$.

Recall that for multisets, $E_H \subseteq E_G$ means for any edge $e, \mu_{E_H}(e) \leq \mu_{E_G}(e)$; in other words, the edge e appears no more often in H than it does in G.

The graph H is a **proper** subgraph of G, written $H < G$, if H is a subgraph of G and $H \neq G$.

I.E.2. INDUCED SUBGRAPHS

If $G = (V_G, E_G)$ is a graph and $W \subseteq V_G$ is given, then let H be the graph with vertex set W and edge multiset E_H where for every edge e with both endpoints in W, $\mu_{E_H}(e) = \mu_{E_G}(e)$. We say that H is the subgraph of G that is **induced** by the set W. We write $G(W)$ for the subgraph of G induced by W.

If $G(W)$ is a complete graph, then it is called a **clique**.

If $G(W)$ is totally disconnected, then W is called an **independent set** of vertices.

Theorem I.E.2.a. If $H \leq G$ and $W \subseteq V_H$, then $H(W) \leq G(W)$.

Proof. Let $H' = H(W)$ and $G' = G(W)$. Their vertex sets are both W by definition, so they are equal. For the edges, we have $\mu_{E_{H'}}(e) = \mu_{E_H}(e) \leq \mu_{E_G}(e) = \mu_{E_{G'}}(e)$, implying the result.

▫

I.F. GRAPH ISOMORPHISM

Given two graphs, we want to clarify when they have identical structures. For instance, any two triangles are simple graphs with three vertices and three edges.

More formally, we develop the machinery of graph isomorphism:

Definition. Given two graphs G and G', a **graph isomorphism** is a bijection $\phi: V_G \to V_{G'}$ such that if $\phi(u) = u'$ and $\phi(v) = v'$, then the multiplicity of uv in E_G is the same as that of $u'v'$ in $E_{G'}$.

Definition. Given two graphs G and G', if there exists a graph isomorphism $\phi: V_G \to V_{G'}$, then G and G' are said to be **isomorphic.**

Observations. If G and G' are isomorphic graphs, then

- They have the same degree sequences where loops contribute 2 to a vertex's degree.
- They are the same order and the same size.
- Either both are bipartite or both fail to be bipartite.
- Either both are simple or both fail to be simple.
- For any subgraph of G, there is an isomorphic subgraph of G'.

I.G. GRAPH OPERATIONS

I.G.1. THE COMPLEMENT OF A SIMPLE GRAPH

Given a simple graph $G = (V, E)$, its **complement** is the simple graph $\overline{G} = (V, \overline{E})$. Notice that the vertex set is the same for the complement while the edge set of $\overline{G}$ is $\overline{E} = \{uv\colon\ u \neq v \text{ and } uv \notin E\}$.

Observations:

- e is an edge of G if and only if e is not an edge of $\overline{G}$,
- For distinct vertices u and v, u is adjacent to v in G if and only if u is not adjacent to v in $\overline{G}$,
- The complement of a complete graph is the totally disconnected graph with the same vertex set.

I.G.2. THE UNION AND THE INTERSECTION

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$,

- Their **union** $G_1 \cup G_2$ has vertex set $V_1 \cup V_2$ and edge multiset $E_1 \cup E_2$. Recall that for multisets, the multiplicity of e in the union is the maximum of the multiplicities in the two sets.
- Their **intersection** $G_1 \cap G_2$ has vertex set $V_1 \cap V_2$ and edge multiset $E_1 \cap E_2$. Recall that for multisets, the multiplicity of e in the union is the minimum of the multiplicities in the two sets. If $V_1 \cap V_2 = \emptyset$, then $G_1 \cap G_2$ is a null graph. If $E_1 \cap E_2 = \emptyset$, then $G_1 \cap G_2$ is an empty graph.

I.G.3. THE DISJOINT UNION

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we define their **disjoint union** $G_1 \sqcup G_2$ as the graph whose vertex set is $V = \{(v, 1)\colon\ v \in V_1\} \cup \{(w, 2)\colon\ w \in V_2\}$ and for each $e = uv \in E_i$ there is a corresponding edge e' of $G_1 \sqcup G_2$ whose endpoints are (u, i) and (v, i). The reason for using ordered pairs to represent vertices is to ensure there is no overlap between the two sets of vertices used to form the union. A common visualization is to use "colors," one for the vertices of G_1 and another for the vertices of G_2.

This generalizes easily to indexed families of graphs $G_i = (V_i, E_i), i \in I$, some index set. For each $i \in I$, we define the set

$$W_i = \{(v, i)\colon\ v \in V_i\}.$$

This has the effect of appending the value i to the vertex v. Then if G is the disjoint union of the G_i graphs, i.e., in symbols,

$$G = \bigsqcup_{i \in I} G_i,$$

the graph G has vertex set $V = \bigcup_{i \in I} W_i$ and for each $e = uv \in E_i$ there is a corresponding edge e' of $G_1 \sqcup G_2$ whose endpoints are (u, i) and (v, i).

I.G.4. THE CARTESIAN PRODUCT

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we define their **Cartesian product** $G_1 \times G_2$ as the graph whose vertex set is $V = V_1 \times V_2$ and where vertices $v = (v_1, v_2)$ and $w = (w_1, w_2)$ are adjacent if either

- $v_1 = w_1$ and v_2 is adjacent to w_2, or
- $v_2 = w_2$ and v_1 is adjacent to w_1.

This generalizes to a family of graphs G_i, $i \in I$. If v and w are I-tuples, then they are adjacent if there exists $k \in I$ such that v_k is adjacent to w_k in the graph G_k and for all indices $j \neq k$, $v_j = w_j$.

If $I = \{1,2,3, \dots, n\}$ and $G_i = G$, the same graph for all indices, then we have the n-fold Cartesian product

$$G^n = \underbrace{G \times G \times \cdots \times G}_{n \text{ times}}.$$

Observations:

- If $|V_1| = n_1$ and $|V_2| = n_2$, then $|V| = |V_1 \times V_2| = n_1 n_2$.
- If $|E_1| = m_1$ and $|E_2| = m_2$, then $|E| = n_1 m_2 + n_2 m_1$.

Example a. Let Q_1 be the 1-cube graph whose vertex set is {0,1} and where 0 is adjacent to 1. Then

$$Q_n = Q_1^n.$$

Example b. Given two path graphs P_k and P_n, the graph $P_k \times P_n$ can be visualized as a **grid** (as in Figure I.F.4.b, showing $P_5 \times P_3$.)

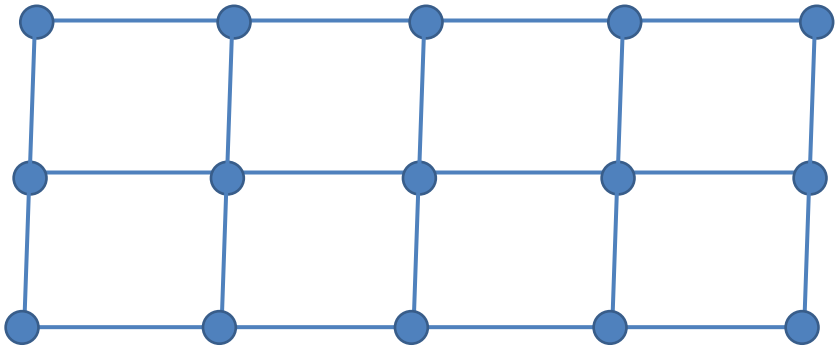


Figure I.F.4.b. The grid graph $P_5 \times P_3$

Higher-dimensional analogs of grid graphs can be constructed by forming Cartesian products of more paths.

Example c. Given two cycle graphs C_k and C_n, the graph $C_k \times C_n$ can be visualized as a **torus** (as in Figure I.F.4.c, showing $C_5 \times C_3$.)

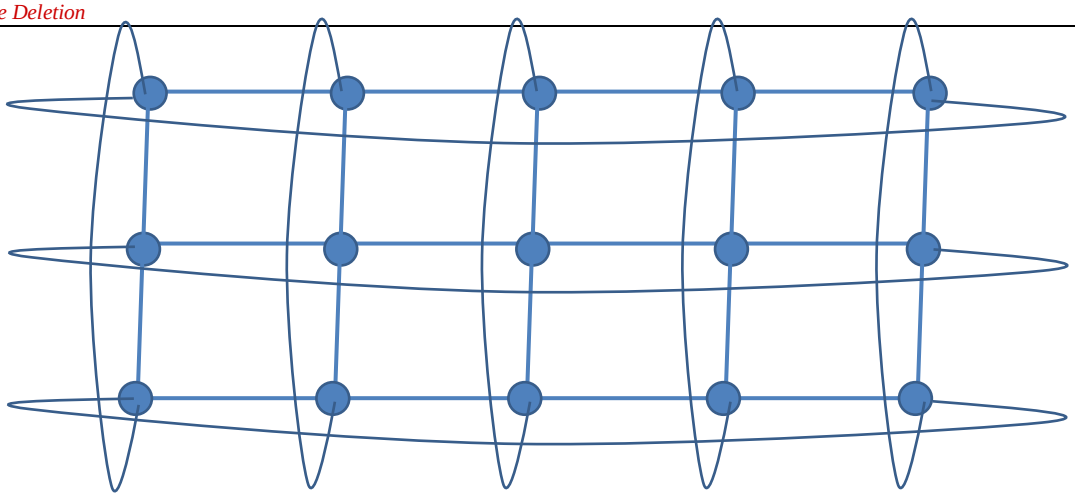


Figure I.F.4.c. The torus graph $C_5 \times C_3$

Observation. If $k, n \geq 3$ then $C_k \times C_n$ is 4-regular.

Higher-dimensional analogs of torus graphs can be constructed by forming Cartesian products of more cycles.

Example d. Let Z be the graph whose vertex set is $\mathbb{Z}$ and where two vertices are adjacent if and only if they differ by 1. The graph Z^n has vertices that are lattice points of $\mathbb{Z}^n$ where two vertices are adjacent if they are exactly 1 unit apart.

I.G.5. EDGE DELETION

Given a graph $G = (V, E)$ and an edge $e \in E$, the graph $G - e$ produced by **deleting the edge e** has vertex set V and edge set $E - \{e\}$; the latter notation is understood to mean the multiplicity of e is reduced by 1 in E.

I.G.6. VERTEX DELETION

Given a graph $G = (V, E)$ and a vertex $v \in V$, the graph $G - v$ produced by **deleting the vertex v** is the induced subgraph $G(V - \{v\})$. This is the subgraph obtained by deleting the vertex v from V and by deleting all edges incident with v from E.

If S is a set of vertices, then $G - S$ is the graph produced by deleting all of the vertices in S.

Example a. A **caterpillar** is a graph G that is either a path or such that if S is the set of all degree-1 vertices, then $G - S$ is a path. See Figure I.G.6.a for an example:

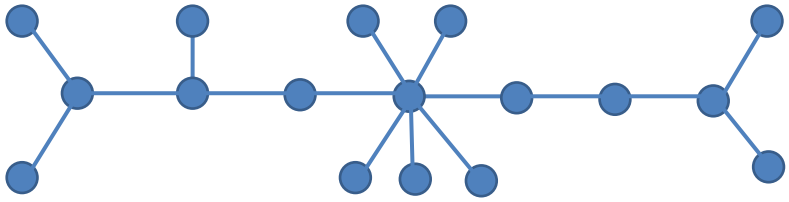


Figure I.G.6.a. A Caterpillar

I.G.7. VERTEX COLLAPSING/IDENTIFICATION; EDGE CONTRACTION

Given a graph $G = (V, E)$ and a set of vertices $W \subseteq V$, the graph $G/W = (V', E')$ obtained by **collapsing the vertices in W** has vertex set $V' = (V - W) \cup \{w\}$. Here, the symbol w is used to represent a new vertex not already in V.

We form E' by the following recipe: For each edge $e = uv \in E$,

- If $u, v \in V - W$, the edge e is in E'.
- If $u \in V - W$ and $v \in W$, e becomes the edge uw in E'. Here, if $u \in W$ and $v \in V - W$, switch the labels u and v.
- If $u, v \in W$, e becomes a loop ww.

Identifying the vertices in W is the same operation except no loops ww are created, nor are parallel edges. This is the typical version of this operation performed on simple graphs, so the result is also a simple graph.

The edge $e = uv$ is **contracted** by identifying the set $W = \{u, v\}$. If e is among parallel copies of itself, all copies are contracted simultaneously. Contracting a loop is the same as deleting it.

I.G.8. VERTEX SPLITTING

Given a graph $G = (V, E)$, a vertex $v \in V$, and a set W that does not intersect V, the graph $G(v \to W)$ obtained by **splitting v into W** has vertex set $V' = (V - \{v\}) \cup W$ and the edge set E' is formed by replacing each edge $e = uv \in E, u \neq v$, with the edges $uw \in E'$ for each $w \in W$. Each loop vv is replaced with the edges $ww' \in E'$ for every $w, w' \in W$.

I.G.9. EDGE SUBDIVISION AND MERGING

Given a graph $G = (V, E)$, an edge $e = uv \in E$, and a positive integer n, the edge e is **subdivided n times** by forming the graph G' whose vertex set is $V' = V \cup \{w_1, w_2, \dots, w_n\}$ and whose edge set E' is formed by the following recipe, beginning with $E' = E$:

- e is deleted from E'
- For each $i, 1 \leq i \leq n-1$, the edge $w_i w_{i+1}$ is inserted in E'
- The edges uw_1 and $w_n v$ are inserted into E'

Visually, this has the effect of "depositing" n new vertices along e. A **subdivision** of a graph G is obtained by subdividing a set of edges of G, each one any number of times. The reverse operation is called **merging**. Given a sequence of edges $uw_1, w_1w_2, w_2w_3, \dots, w_k v$ where the vertices w_i are all distinct with degree 2, then we **merge** the edges by deleting the vertices w_i and inserting the edge uv into E. Figure I.G.9.a. shows a subdivision of K_4.

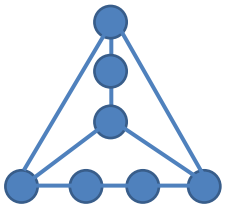


Figure I.G.9.a. A Subdivision of K_4

SECTION II. CONNECTIVITY

II.A. WALKS AND THEIR REFINEMENTS

II.A.1. WALKS

Definition. Given a graph $G = (V, E)$, a **walk** W is a finite sequence alternating between vertices and edges that begins and ends at vertices, i.e.,

$$W: v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$$

and such that for all $k, 1 \leq k \leq n$, $e_k = v_{k-1}v_k$; also, if $v_{k-1} = v_k$, then e_k is a loop joining v_k to itself.

- The graph G is said to **admit** the walk W.
- A **trivial walk** is a sequence having a single vertex and no edges.
- The vertices v_0 and v_n are the **starting point** and **ending point** of W. The remaining vertices, including repetitions of v_0 or v_n are called **interior points** of W.
- The integer n is called the **length** of the walk, denoted $|W|$. This is the number of edges listed in the walk. A trivial walk has length 0.
- If $|W| < |W'|$, then W is **shorter** than W'. If $|W| > |W'|$, then W is **longer** than W'.
- Given vertices u and v, a u, v-**walk** has u as its starting point and v as its ending point.
- The **reverse walk** to $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ is $v_n, e_n, v_{n-1}, \dots, v_2, e_2, v_1, e_1, v_0$. This satisfies the definition of being a walk.
- Given two walks $W: v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ and $W': v_n, e_{n+1}, v_{n+1}, \dots, v_{p-1}, e_p, v_p$, they can be **concatenated** to form the walk $WW': v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n, e_{n+1}, v_{n+1}, \dots v_{p-1}, e_p, v_p$.

It is common to interpret a walk as visiting the vertices one at a time. The vertices are **encountered** in the order $v_0, v_1, \dots, v_n$. The edges are **traversed** in the order $e_1, e_2, \dots, e_n$.

If G is a graph without parallel edges, then one can specify a walk by listing the vertices as they are encountered.

II.A.2. PATHS AND TRAILS

A **path** $P: v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$ is a walk that does not repeat a vertex.

A **Hamilton path** is a path that includes every vertex of the graph.

Theorem II.A.2.a. If a graph G admits a u, v-walk that repeats a vertex, then G admits a shorter u, v-walk.

Proof. Suppose G admits a u, v-walk $W: v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$ that repeats a vertex. Let $i < j$ be given such that $v_i = v_j$. Then, if $j < k$, the walk

$$W': v_0, e_1, v_1, e_2, v_2, \dots, e_i, v_i, e_{j+1}, v_{j+1}, \dots, e_k, v_k$$

has length $|W'| = k - (j - i) < k = |W|$. If $j = k$, then the walk

$$W': v_0, e_1, v_1, e_2, v_2, \dots, e_i, v_i$$

has length $|W'| = i < j = k = |W|$. In either case, $|W'| < |W|$.

□

Corollary II.A.2.b. Any shortest u, v-walk is a path.

A **trail** is a walk that does not repeat an edge.

An **Euler trail** is a trail that includes every edge of the graph.

II.A.3. DISTANCE BETWEEN VERTICES

Given two vertices u and v, it is useful to have a measure of distance between them.

Definition. The **distance** between vertices u and v, denoted $D(u, v)$, is the length of a shortest u, v-walk. If there is no such walk, then $D(u, v) = \infty$.

Notice that a trivial u, u-walk has length zero, so $D(u, u) = 0$ for any vertex u.

In fact, the distance D satisfies the usual requirements for an **extended** (allows for ∞) **metric**:

- For any vertex u, $D(u, u) = 0$.
- For any vertices u and v, if $u \neq v$, then $D(u, v) > 0$.
- For any vertices u and v, $D(u, v) = D(v, u)$. This follows from the fact that reversing a u, v-walk of minimum length results in a v, u-walk of minimum length.
- For any vertices u, v, w, $D(u, w) \leq D(u, v) + D(v, w)$. Here, if P is a minimum length u, v-walk and Q is a minimum length v, w-walk, then their concatenation PQ is a u, w-walk whose length is the sum of the lengths of P and Q. It is possible that PQ could be shortened, so that there could be a shorter u, w-walk.

Theorem II.A.3.a. If P is a shortest u, v-path and x is a vertex on P, then the u, x-subpath along P is a shortest u, x-path and the x, v-subpath along P is a shortest x, v-path.

Proof. Let P' be the u, x-subpath along P and Q' be the remaining x, v-subpath along P. This means $P = P'Q'$. If P' is not a shortest u, x-subpath, then there exists some shorter u, x-walk P''. But then $P''Q'$ is a shorter u, v-walk than P, which cannot happen. Similarly for Q'.

□

II.A.4. CLOSED WALKS, CIRCUITS, AND CYCLES

A **closed walk** is a walk whose starting and ending points are the same vertex, i.e., if $v_0 = v_n$, then the walk

$$W\colon\ v_0, e_1, v_1, \ldots, e_n, v_n$$

is a closed walk. A trivial walk is considered a closed walk.

A **circuit** is a closed walk that is also a trail.

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

A **cycle** is a circuit with at least one edge where the only vertex that appears more than once occurs at the starting and ending points.

II.A.5. EVEN CYCLES AND BIPARTITE GRAPHS

We develop a straightforward criterion based on cycle lengths for a graph to be bipartite.

Definition. A cycle C_n is an **even cycle** if n is even and an **odd cycle** if n is odd.

Some warm-up results:

Lemma II.A.5.a. Odd cycles are not bipartite.

Proof. Let

$$C_{2n+1}: v_0, e_1, v_1, \dots, v_{2n}, e_{2n+1}, v_{2n+1} = v_0$$

be an odd cycle. If this were a bipartite graph, we could partition the vertices into two parts V_0, V_1 such that every edge has one endpoint in V_0 and the other in V_1. Suppose this happens and let V_0 be the part that contains vertex v_0. We claim that this implies $v_{2k} \in V_0$ and $v_{2k+1} \in V_1$ for all $k, 0 \leq k \leq n$. Proceeding inductively, we already have $v_0 \in V_0$ by hypothesis. Also, since $e_1 = v_0v_1$, this implies $v_1 \in V_1$. Assume the result holds for $k = i$, for a given $i, 0 \leq i \leq n-1$; i.e., we assume $v_{2i} \in V_0$ and $v_{2i+1} \in V_1$. Then, since $e_{2i+2} = v_{2i+1}v_{2i+2}$, we have $v_{2i+2} \in V_0$. Also, since $e_{2i+3} = v_{2i+2}v_{2i+3}$, we have $v_{2i+3} \in V_1$, which completes the inductive step. Thus, $v_0 \in V_0$ and $v_0 = v_{2n+1} \in V_1$ which implies $V_0 \cap V_1 \neq \emptyset$, contradicting the fact that V_0 and V_1 partition V. Hence, C_{2n+1} is not bipartite.

□

Lemma II.A.5.b. Every subgraph of a bipartite graph is bipartite.

Proof. Let G be bipartite with parts U and W. Also, let $G' \leq G$ be given with vertex set V'. We let $U' = V' \cap U$ and $W' = V' \cap W$. At once, we have

$$V' = V' \cap V = V' \cap (U \sqcup W) = (V' \cap U) \sqcup (V' \cap W) = U' \sqcup W'$$

and so V' is partitioned into U' and W'. Also, if $e = xy$ is an edge in G', then both x and y belong to V'. Since e is also an edge in G, without loss of generality, we have $x \in U$ and $y \in W$. Thus, $x \in V' \cap U = U'$ and $y \in V' \cap W = W'$, and so G' is bipartite with parts U' and W'.

□

Corollary II.A.5.c. If G is a bipartite graph, then G contains no odd cycles as subgraphs.

Lemma II.A.5.d. Any arbitrary disjoint union of bipartite graphs is bipartite.

Proof. Let I be an index set and for each $i \in I$, let G_i be a bipartite graph with parts U_i and W_i; we assume the vertex sets V_i are pairwise disjoint (if not, we form the disjoint union using the construction in I.G.3). This implies $U_i \cap W_j = \emptyset$ for all i, j. Consider the unions $V = \bigcup_{i \in I} V_i$, $U = \bigcup_{i \in I} U_i$ and $W = \bigcup_{i \in I} W_i$. It is routine (Left to Reader) to show $U \cap W = \emptyset$ and $U \cup W = V$. For any edge e in $G = \bigsqcup_{i \in I} G_i$, there is a

unique $i \in I$ such that $e = xy, x \in U_i, y \in W_i$. This implies $x \in U$ and $y \in W$ and so G satisfies the requirements for being a bipartite graph.

The main result in this section is the converse:

Theorem II.A.5.e. If G has no odd cycles as subgraphs, then G is bipartite.

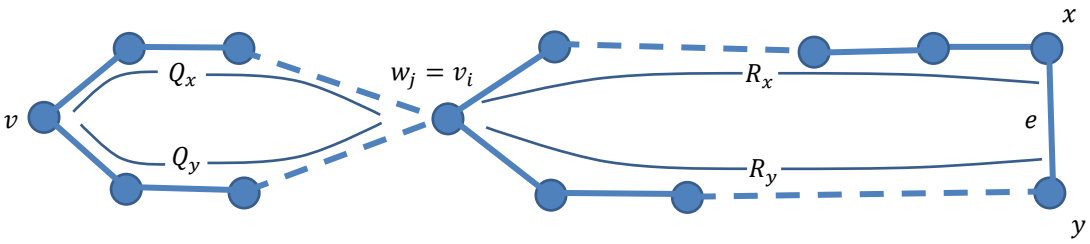
Proof. We prove the result for connected graphs, G has no odd cycles as subgraphs. If G is not connected, we can use this result on each component and then Lemma II.A.5.d for the disjoint union of the components. We want to show that we can partition $V = V_0 \sqcup V_1$ so that every edge has an endpoint in V_0 and the other endpoint in V_1. Choose a vertex v in G. We define V_0 and V_1 as follows:

$$V_0 = \{w \in V\colon\ D(v,w) \text{ is even}\}$$
$$V_1 = \{w \in V\colon D(v,w) \text{ is odd}\}$$

It is immediate that $V = V_0 \sqcup V_1$. Now, let $e = xy$ be any edge. Suppose $x \in V_0$ and $y \in V_0$. Then there are even-length shortest v,x- and v,y-paths

$$P_x\colon\ v = v_0, e_1, v_1, \dots, v_{2p-1}, e_{2p}, v_{2p} = x$$
$$P_y\colon v = w_0, f_1, w_1, \dots, w_{2q-1}, f_{2q}, w_{2q} = y$$

Suppose w_j is the highest-indexed vertex on P_y that is also a vertex v_i on P_x. We note that $w_0 = v_0$, so j exists.



We note that none of the v_k or w_ℓ vertices can be equal if $k > i$ or $\ell > j$.

Let Q_x be the v, v_i-subpath of P_x and R_x be the v_i, x-subpath of P_x. Also, let Q_y be the v, w_j-subpath of P_y and R_y be the w_j, y-subpath of P_y. In short, $P_x = Q_x R_x$ and $P_y = Q_y R_y$. If $j < i$, then $Q_y R_x$ is a shorter v,x-walk than $P_x = Q_x R_x$, which contradicts P_x being shortest. If $i < j$, then $Q_x R_y$ is a shorter v,y-walk than $P_y = Q_y R_y$, which contradicts P_y being shortest. Hence, $i = j$. It follows that

$$C\colon v_i, e_{i+1}, v_{i+1}, \dots, v_{2p} = x, e, y, w_{2q}, f_{2q}, w_{2q-1}, \dots, w_i = v_i$$

is a cycle of length $(2p - i) + 1 + (2q - i) = 2(p + q - i) + 1$ which is an odd number. Hence, C is an odd cycle, contradicting G having no odd cycles.

A similar argument works if x and y both belong to V_1. This implies x and y must belong to distinct V_0, V_1 sets. This holds for every edge, and so G is bipartite.

□

II.B. CONNECTEDNESS AND CONNECTIVITY

The title may seem redundant, but we reserve "connectedness" to be a condition that is either held or not held and "connectivity" to be a numerical value that indicates how strongly a graph is connected, i.e., the higher the connectivity, the harder it is to disconnect a graph by deleting vertices or edges.

II.B.1. VERTEX CONNECTEDNESS

Given a graph G, two vertices $u, v \in V$ are **connected** if there exists a u, v-walk. We write $u \sim v$ when this occurs.

Observations.

- There exists a u, v-walk if and only if there exists a u, v-path.
- Since the trivial u, u-walk always exists, a vertex u is always connected to itself. Hence, for all vertices $u \in V$, $u \sim u$.
- For any vertices u and v, if $u \sim v$, then we can reverse any existing u, v-walk to produce a v, u-walk. Hence, $v \sim u$.
- For any vertices u, v, and w, if $u \sim v$ and $v \sim w$, then we can concatenate any existing u, v-walk with any existing v, w-walk to produce a u, w-walk. Hence, $u \sim w$.

The last three observations establish the following result.

Theorem II.B.1.a. Vertex connectivity is an equivalence relation over V.

Notation. Given a vertex v of a graph G, the set $\{u \in V: u \sim v\}$ of vertices connected to v is denoted $[v]$. This is an equivalence class.

II.B.2. GRAPH CONNECTEDNESS AND COMPONENTS

Definition. A non-null graph G is **connected** if for any two vertices $u, v \in V$, u and v are connected. Intuitively, one can reach any vertex from any other vertex by traversing edges.

Definition. The **component** of a graph G that contains v is the induced subgraph $G([v])$ where $[v] = \{u \in V: u \sim v\}$ is the equivalence class of vertices that are connected to v. Intuitively, this is the largest "connected piece" of G that contains v.

Conventions and Observations.

- Convention: A null graph is not connected.
- Convention: A null graph has zero components.
- A component is connected.
- A trivial graph has one component.
- A connected graph has one component.
- For any two vertices $u, v \in V$, the following are equivalent:
  - u is connected to v,
  - $u \in [v]$,

- $[u] \cap [v] \neq \emptyset$,
- $[u] = [v]$,
- $G([u]) = G([v])$,
- u and v are vertices in the same component.

- Every vertex belongs to exactly one component.
- Every edge belongs to exactly one component. Otherwise, the endpoints of a counterexample would not belong to exactly one component.

We establish some preliminary results that will lead to a fundamental result concerning edges and components.

Theorem II.B.2.a. For any vertices u, v in the same component of G, every edge in any u,v-walk is an edge of $G([u])$.

Proof. Suppose $e = st$ is an edge in a u,v-walk W. Then both s and t are encountered on W as e is traversed, i.e., either s, e, t or t, e, s is a subsequence in W. This implies s and t are vertices in W and so there exist u,s-walks and u,t-walks. Thus, $u \sim s$ and $u \sim t$ and so $s \in [u]$ and $t \in [u]$, implying $e = st$ is an edge of $G([u])$.

□

Theorem II.B.2.b. If H is a subgraph of G and $u, v \in V(H)$ are in the same component of H, then they are in the same component of G.

Proof. Let W be a u,v-walk of H. Since $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, the walk W is a u,v-walk of G.

□

Corollary II.B.2.c. If H is a subgraph of G and $u, v \in V(H)$ are in different components of G, then they are in different components of H.

II.B.3. BRIDGES

We now focus on how edges influence connectivity.

Definition. Given a graph $G = (V, E)$, an edge $e \in E$ is a **bridge** if there exist vertices u and v such that u and v are connected in G, but not connected in $G - e$.

Intuitively, a bridge is an edge whose removal causes a pair of connected vertices to become disconnected.

Observation. Given a connected graph G and a bridge e of G, u and v are connected in $G - e$ iff there exists a u,v-walk in G without e.

□

Rephrasing this in contrapositive form, we have

Theorem II.B.3.a. Given a connected graph G and a bridge e of G, u and v are disconnected in $G - e$ iff every u, v-walk in G contains e.

Theorem II.B.3.b. Given a connected graph G and a bridge e of G, deleting e disconnects its endpoints.

Proof. Let $e = st$ and suppose u and v become disconnected in $G - e$. If s and t are connected in $G - e$, then there exists an s, t-path P in $G - e$. Let Q be a u, v-path in G:

$$Q: u = v_0, e_1, v_1, \dots, v_i = s, e = e_i, t = v_{i+1}, e_{i+1}, \dots, v_n = v.$$

Then

$$Q': u = v_0, e_1, v_1, \dots, e_i, P, e_{i+1}, \dots, v_n = v$$

is a u, v-walk in $G - e$ which contradicts u and v becoming disconnected in $G - e$.

□

Corollary II.B.3.c. Neither loops nor one of a set of parallel edges are bridges.

Theorem II.B.3.d. Given a connected graph G and a bridge $e = st$ of G, $V = [s]_{G-e} \sqcup [t]_{G-e}$.

Proof. We first observe that since s and t are disconnected in $G - e$, $[s]_{G-e} \cap [t]_{G-e} = \emptyset$, justifying the use of the disjoint union symbol for the right-hand side. The inclusion $[s]_{G-e} \sqcup [t]_{G-e} \subseteq V$ is immediate.

To show $V \subseteq [s]_{G-e} \sqcup [t]_{G-e}$, we assume $w \in V$ and let P, Q be s, w- and t, w- paths in G. If e is not on one of these paths, then $w \in [s]_{G-e}$ or $w \in [t]_{G-e}$. If e is on $P: s = v_0, e_1, v_1, e_2, v_2, \dots, w$, then $e_1 = e$ and $v_1 = t$ since otherwise s would be a repeated vertex. But then $P^*: t = v_1, e_2, v_2, \dots, w$ is a t, w-path in $G - e$ and so $w \in [t]_{G-e}$. By switching the roles of s and t, if e is on Q, then $w \in [s]_{G-e}$.

□

Corollary II.B.3.e. Deleting a bridge from a connected graph results in a graph with two components.

Proof. The components are $G([s])$ and $G([t])$ where $[s]$ and $[t]$ are as in the proof of the theorem.

□

Theorem II.B.3.f. Given a graph G and a bridge $e = st$ of G, if $w \notin [s]_G$, then $[w]_{G-e} = [w]_G$.

Proof. The inclusion $[w]_{G-e} \subseteq [w]_G$ is immediate since any w, x-path in $G - e$ is a w, x-path in G.

For the other inclusion, let $x \in [w]_G$ be given. Since $[w]_G \neq [s]_G$ and e is an edge in $G([s]_G)$, e is not an edge in $G([w]_G)$. This implies that any w, x-path will not contain e and so $x \in [w]_{G-e}$.

□

The following two innocent looking results have far-reaching consequences.

Theorem II.B.3.g. Given a graph G with finitely many components, the number of components increases by exactly 1 by deleting a bridge.

Proof. Let $e = st$ be a bridge. From Theorem II.B.3.b,

$$[s]_G = [s]_{G-e} \sqcup [t]_{G-e}.$$

From Theorem II.B.3.d, for $[w]_G \neq [s]_G$, $[w]_G = [w]_{G-e}$. Thus, the number of components of G not containing s is unchanged by deleting e, while the component containing s is split into two components by deleting e. Thus, the number of components increased by exactly one by deleting a bridge.

□

Theorem II.B.3.h. Given a graph G and an edge e of G, e is on a cycle iff e is not a bridge.

Proof. For the forward direction, let

$$C: s = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n = t, e, s$$

be a cycle containing e. Without loss of generality, we can reorient the cycle so that e is the last edge traversed and s is the starting and ending points of C. But then

$$P: s = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n = t$$

is an s,t-path not containing e and so deleting e does not disconnect its endpoints. Hence, e is not a bridge.

For the reverse direction, suppose $e = st$ is not a bridge. Then there exists an s,t-path

$$P: s = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n = t$$

not containing e. But then

$$C: s = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n = t, e, s$$

is a cycle containing e.

□

II.B.4. CUTTING VERTICES AND BLOCKS

Turning our attention to vertices whose removal affects the connectedness of a graph, we have

Definition. Given a graph G, a vertex v is a **cut vertex** if $G - v$ has more components than G.

Deleting a cut vertex is not guaranteed to increase the number of components by any given quantity.
Example. If $k \geq 2$, then deleting the center vertex of a k-star (see Figure II.B.4.a illustrating the case $k = 5$) increases the number of components from 1 to k.

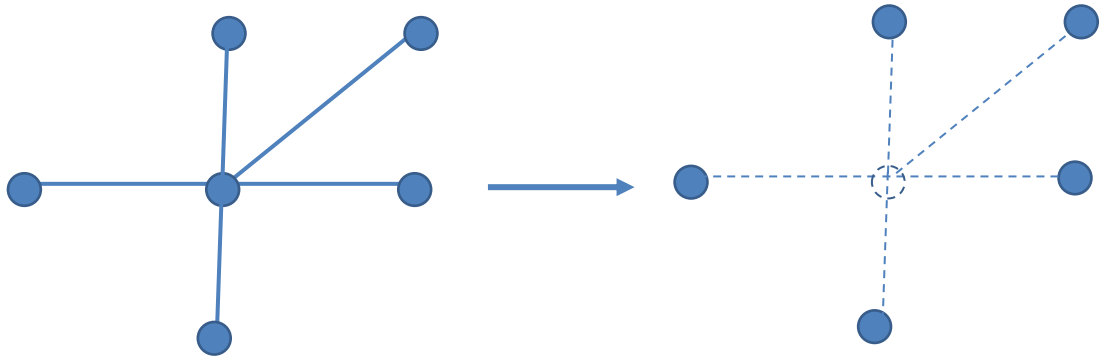


Figure II.B.4.a. Deleting the center vertex of a 5-star

Definitions.

- A graph is **separable** if it is not connected or if it has at least one cut vertex. Otherwise, the graph is **inseparable.**
- If $H \leq G$ is inseparable and there is no other inseparable subgraph $K, H < K \leq G$, then H is a **block** of G.

Observations.

- An inseparable component is a block.
- An isolated vertex is not a cut vertex.

Theorem II.B.4.b. A vertex v is a cut vertex of a connected graph G iff there exist two other distinct vertices u and w of G such that v is on every u,w-path.

Proof. For the forward direction, if v is a cut vertex, then $G - v$ is disconnected with at least two components. Let u and w be any two vertices, each in a distinct component of $G - v$. Then there are no u,w-paths in $G - v$, implying every u,w-path in G must contain v.

For the reverse direction, if every u,w-path in G must contain v, then there is no u,w-path in $G - v$ and so u and w are disconnected. Hence, $G - v$ is disconnected and v is a cut vertex of G.

□

Example II.B.4.c. A **cactus** is a graph G such that every edge is on at most one cycle. A specific type of cactus is called a **Dutch windmill**, where a collection of cycles share a single common vertex. See Figure II.B.4.d for examples of a cactus and a Dutch windmill.

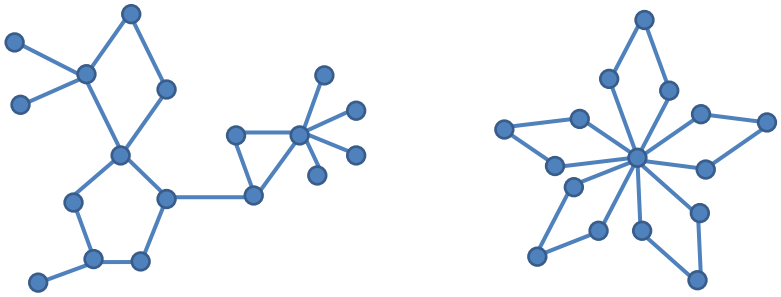


Figure II.B.4.d. A cactus and a Dutch windmill

II.B.5. CUTS AND CUT SETS

We consider two generalizations of bridges, i.e., sets of edges whose removal separates vertices in a graph.

First, suppose we partition the vertex set V into two nonempty parts V_1 and V_2 and then we want to separate the vertices in V_1 from those in V_2. This is accomplished by deleting the edges having one endpoint in V_1 and the other in V_2:

Definition. A **cut** is a partition of V into two nonempty parts V_1 and V_2.

Definition. Given a partition of V into two nonempty parts V_1 and V_2, the **cut induced by V_1 and V_2** is the set of edges $\text{cut}\langle V_1, V_2\rangle = \{e = v_1v_2 \in E: v_1 \in V_1 \text{ and } v_2 \in V_2\}$.

Deleting the edges in $\text{cut}\langle V_1, V_2\rangle$ is necessary and sufficient to separate the vertices in V_1 from those in V_2.

Observation. The graph G is bipartite with parts V_1 and V_2 iff $\text{cut}\langle V_1, V_2\rangle = E$.

Observation. If $|V| \leq 1$, then G has no cuts.

Theorem II.B.5.a. If $\text{cut}\langle V_1, V_2\rangle = \emptyset$ then no vertex in V_1 is in the same component as any vertex in V_2.

Proof. Suppose otherwise, i.e., suppose $u \in V_1$ and $w \in V_2$ are in the same component. Then there is a u,w-path

$$P\colon\ u = v_0, e_1, v_1, \dots, v_{n-1}, e_n, v_n = w.$$

Let $k = \max\{i\colon\ v_i \in V_1\}$. We note that $1 \leq k < n$ and so $k + 1 \leq n$ and $v_{k+1} \notin V_1$, implying $v_{k+1} \in V_2$. But then e_{k+1} has one endpoint $v_k \in V_1$ and the other endpoint $v_{k+1} \in V_2$, implying $e_{k+1} \in \text{cut}\langle V_1, V_2\rangle$, contradicting the hypothesis.

□

Definition. A **cut set** of a connected graph G is a set of edges $F \subseteq E$ such that $G - F$ is disconnected and for any proper subset $H \subset F$, $G - H$ is connected.

Theorem. If F is a cut set for the connected graph G, then $G - F$ has two components.

Proof. Pick an edge $e \in F$. Then if $H = F - \{e\}$, the graph $G - H$ is connected and the graph $(G - H) - e = G - F$ is disconnected. This implies e is a bridge of $G - H$ and so $G - F$ has two components.

□

Theorem. Suppose G is a connected graph. The set $S = \text{cut}\langle V_1, V_2\rangle$ is a cut set iff $G - S$ has exactly two components.

Proof. The forward direction is a direct consequence of the previous theorem. For the reverse direction, we note that $G - S = G(V_1) \cup G(V_2)$ and we note that $G(V_1)$ is disconnected from $G(V_2)$. If there are exactly two components, then they must be $G(V_1)$ and $G(V_2)$. If $H \subset S$, then for every edge e in $S - H$ has one endpoint in V_1 and the other in V_2. But then in the graph $G - H$, every vertex in V_1 is connected to every vertex in V_2. Since every pair of vertices in V_1 are connected and every pair of vertices in V_2 are connected in the graph $G - H$, $G - H$ is connected and so S satisfies the requirements for being a cut set.

□

II.B.6. EDGE- AND VERTEX-CONNECTIVITY

The discussion of cuts and cut sets leads directly to "edge-connectivity," a quantitative measure of how strongly a graph is connected in terms of edges.

Definition. A graph G is said to be ***k*-edge connected** if for any set F of fewer than k edges, $G - F$ is connected.

Definition. The **edge-connectivity** of G, denoted $\lambda(G)$, is the maximum k such that G is k-edge connected.

Observations.

- The minimum cardinality of a cut set is $\lambda(G)$.
- Every graph is 0-edge connected.
- Every connected graph is 1-edge connected.
- Every bridgeless connected graph is 2-edge connected.
- If a graph is k-edge connected, then it is ℓ-edge connected for all $0 \leq \ell \leq k$.
- A graph is k-edge connected for every k, $0 \leq k \leq \lambda(G)$.

Theorem. For any loopless graph G, $0 \leq \lambda(G) \leq \delta(G)$.

Proof. Let v be a vertex of degree $\delta(G)$ and let F be the set of edges incident with v. Then $G - F$ is disconnected and so $G - F$ is not $(\delta(G) + 1)$-connected. The result follows. ▫

The same notions carry over (with a minor twist involving complete graphs) for vertex-connectivity (many authors drop the "vertex" from this phrase and declare a graph's "connectivity" to be attached to this context.)

Definition. A graph G is said to be ***k*-vertex connected** (or k-connected) if it has more than k vertices and for any set W of fewer than k vertices, $G - W$ is connected.

Definition. The **vertex-connectivity** of G, denoted $\kappa(G)$, is the maximum k such that G is k-vertex connected.

Observations.

- A connected graph is 1-connected.
- For a complete graph G on n vertices, we have $\kappa(G) = n - 1$.
- If $\kappa(G) = k$, then there exists a set W of k vertices such that $G - W$ is either disconnected or trivial. The latter condition occurs when G is a complete graph.

Theorem. For any graph G with at least two vertices, $\kappa(G) \leq \lambda(G)$.

Proof. We can assume WLOG that G has no loops; deleting loops has no effect on connectivity. Also, if G is not connected, then both quantities are zero. Thus, we may assume G is connected. Let $k = \lambda(G)$ and let $F = \{s_1t_1, s_2t_2, \ldots, s_kt_k\}$ be a set of k edges where $G - F$ is disconnected (with two components) and where the s_i vertices are in the same component K of $G - F$. Notice that the t_j vertices are in the other

component L of $G - F$. If $G - \{s_1, s_2, \dots, s_k\}$ is disconnected, then $\kappa(G) \leq k = \lambda(G)$. Otherwise, K has no vertices other than $s_1, s_2, \dots, s_k$. In this case, consider a neighbor w of s_1 in G. First, every neighbor of s_1 takes the form s_i or t_j. If $s_j = s_1$, then $w = t_j$. If $s_j \neq s_1$, then there is an edge $s_1 t_j$ and then t_j is also in K, contradicting t_j being a vertex in L. Thus, the neighbors of s_1 are partitioned into two sets:

$$N(s_1) = \{t_j\colon\ s_j = s_1\} \sqcup \{s_i\colon\ s_i \neq s_1 \text{ and } s_1 s_i \in E\}$$

Thus,

$$|N(s_1)| = \left|\{t_j\colon\ s_j = s_1\}\right| + |\{s_i\colon\ s_i \neq s_1 \text{ and } s_1 s_i \in E\}| \leq k.$$

Finally, if we delete the vertices in $N(s_1)$, we separate s_1 from the remaining vertices, or we leave s_1 as the only vertex in a trivial graph, and so $\kappa(G) \leq k$, as desired.

□

II.C. TRAVERSALS

We turn to considering graph traversals, i.e., walks that "cover" a finite graph in some prescribed way. The first of these traversal schemes, named for Euler, addresses when it is possible to construct a walk or a circuit that uses every edge exactly once. The second scheme, named for Hamilton, addresses when it is possible to construct a path or cycle that uses every vertex exactly once.

II.C.1. EULERIAN GRAPHS

The famous "Bridges of Königsberg" problem is generally credited as the problem that gave birth to graph theory. The story goes that the city of Königsberg (at the time, in Prussia; today, this city is Kaliningrad, Russia) was split by the river Preger into four land masses. Seven bridges were used to connect the land masses to one another, as sketched in Figure II.C.1.a.

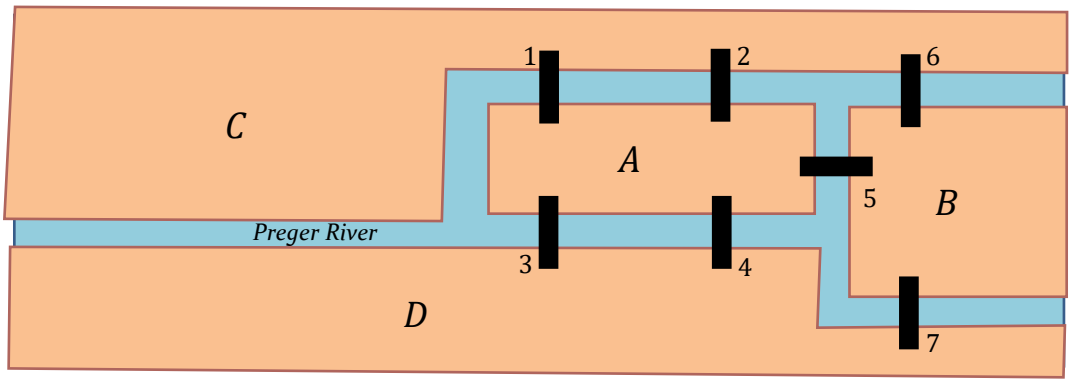


Figure II.C.1.a. Schematic of Königsberg's seven bridges

The legend goes that the citizens of Königsberg tried but failed to find a route through the city that used each bridge exactly once (without swimming in the river). Königsberg was a university city with a vibrant mathematical community and so it was no accident that Leonhard Euler was a frequent visitor. Euler was able to show that no route was possible. His inspiration was to represent the land masses as vertices (vertex set $V = \{A, B, C, D\}$) and the bridges as edges (edge set $E = \{1,2,3,4,5,6,7\}$) of a graph (possibly the first such representation, see Figure II.C.1.b.)

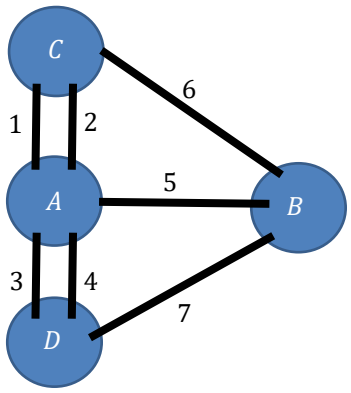


Figure II.C.1.b. Euler's graph representation of the bridges of Königsberg

Euler's argument of the impossibility of a walk using each bridge once can be summarized in graph theoretical language.

Proposition II.C.1.c. A Königsberg walk using each edge exactly once is impossible.

Proof. Suppose otherwise and let

$$W\colon\ v_0, e_0, v_1, e_1, \dots, e_6, v_6$$

be a Königsberg walk, where each $v_i \in \{A, B, C, D\}$ and each $e_j \in \{1,2,3,4,5,6,7\}$. We note that each bridge is used exactly once, so the W has length 7. For every interior vertex v_1 through v_5, two distinct edges are incident. For the end vertices, one edge is incident. So the degree of a given A, B, C, D is twice the number of times it is an interior vertex added to once the number of times it is an end vertex. Thus, if the degree is odd, it must be an end vertex. But all four vertices have odd degree and so there must be four end vertices. This is a contradiction because there are only two end vertices.

□

Definition. An **Euler walk** is a walk in a finite graph that contains a unique instance of every edge in the graph.

Definition. An **Euler circuit** is an Euler walk that starts and ends at the same vertex. A graph with an Euler circuit is said to be **Eulerian.**

Lemma II.C.1.d. A finite graph with minimum degree at least 2 has a cycle as a subgraph.

Proof. Let G be a graph whose minimum degree is at least 2. We form a walk W starting at any vertex v_0. Since v_0 is not isolated, there is an edge e_1 not yet used in W such that e_1 incident with v_0 and its other endpoint is v_1. If $v_1 = v_0$, then W is a 1-cycle v_0, e_1, v_0 which we can use as our subgraph. If $v_1 \neq v_0$, then since v_1 has degree at least 2, there is an edge $e_2 \neq e_1$ incident with v_1; we assign v_2 to be the other endpoint of e_2. More generally, suppose we have constructed

$$W\colon\ v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$$

where all edges are distinct and all vertices except possibly v_k are distinct. If $v_k = v_i$ for some $i, 0 \leq i < k$, then

$$C\colon\ v_i, e_{i+1}, v_{i+1}, \dots, v_{k-1}, e_k, v_k = v_i$$

is our cycle. If v_k is distinct from all other v_i, then since $\deg(v_k) \geq 2$, there is an unused edge e_{k+1} incident with v_k and we assign v_{k+1} to be the other endpoint, extending W to include e_{k+1} and v_{k+1}, accordingly. This process must halt within $|V|$ iterations as some vertex will be duplicated by that point. The only way the process halts is by duplication of a vertex at which point a cycle has been constructed.

□

Theorem II.C.1.e. A connected graph has an Euler circuit if and only if every vertex has even degree, where a loop contributes two to the degree of its endpoint.

Proof. We assume G is connected.

For the forward direction, let G have an Euler circuit

$$S\colon\ v_0, e_1, v_1, e_2, v_2, \dots, v_{m-1}, e_m, v_m = v_0.$$

Except for v_0, the degree of each vertex v is twice the number of times it is encountered on the walk. This is a consequence of each occurrence of v being incident on two edges and there being no loops. The vertex v_0 has a degree twice the number of times it occurs as an interior vertex and added to 2, accounting for the edges e_1 and e_m, also incident with v_0. All of these degrees are even numbers.

For the reverse direction, suppose counterexamples exist. We note that if G is trivial, the trivial circuit is an Euler circuit, so this is not a counterexample. Let G be a counterexample with the minimum number of edges m. The graph G must be connected with every vertex having even degree; since G is nontrivial, each vertex must have degree at least 2. By lemma II.C.1.d, there exists a subgraph C that is a cycle. We suppose

$$C\colon\ v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k = v_0$$

is the cycle.

Let $H = G - E(C)$, the graph obtained by deleting the edges of C. If $v \neq v_i$ for any i, then $\deg_H(v) = \deg_G(v)$, an even number. If $v = v_i$, then $\deg_H(v) = \deg_G(v) - 2$, also an even number. Thus, every vertex of H has even degree. This implies every vertex of every component of H has even degree. We number these components $H_1, H_2, \dots, H_p$ in the order they are first encountered as we traverse the vertices in C in their order. Since H_i has fewer edges than G, the component is not a counterexample and so it has an Euler circuit

$$C_i\colon\ w_{i0}, e_{i1}, w_{i1}, \dots, w_{i(p-1)}, e_{ip}, w_{ip} = w_{i0}$$

where w_{i0} is the first vertex v_j in C contained in H_i. If we then replace each v_j with C_i (not replacing the final occurrence of v_0) for all components H_i, we obtain an Euler circuit for G. This contradicts G being a counterexample, and so no counterexample exists. The result follows.

□

Corollary II.C.1.f. A connected graph has an Euler walk that is not a circuit if and only if it has exactly two vertices of odd degree.

Proof. For the forward direction, suppose G has an Euler walk

$$W\colon\ v_0, e_1, v_1, \dots, v_{m-1}, e_m, v_m.$$

Then let e_{m+1} be a new edge joining v_0 and v_m. The graph $G + e_{m+1}$ then has an Euler circuit

$$S\colon v_0, e_1, v_1, \dots, v_{m-1}, e_m, v_m, e_{m+1}, v_0.$$

This implies every vertex of $G + e_{m+1}$ has even degree and so in the graph $G = (G + e_{m+1}) - e_{m+1}$, the vertices v_0 and v_m have odd degree and all of the other vertices have even degree.

For the reverse direction, let G be connected with exactly two vertices u, w of odd degree. Then $G + uw$ is connected with every vertex having even degree. This latter graph has an Euler circuit. Deleting uw from this circuit results in an Euler u,w-walk.

II.C.2. HAMILTONIAN GRAPHS

We now consider walks that visit every vertex exactly once (this must be a path), or repeats only at the end vertices (this must be a cycle.) Here, the adjectives "Hamilton" or "Hamiltonian" describes this property.

Definition. A **Hamilton path** in a graph G is a path subgraph that contains every vertex.

Definition. A **Hamilton cycle** in a graph G is a cycle subgraph that contains every vertex. A graph that has a Hamilton cycle is said to be **Hamiltonian**.

Observation. Every Hamiltonian graph has a Hamilton path; simply remove an edge from a Hamilton cycle to produce a Hamilton path.

Unlike the case with Eulerian circuits or walks, there are no straightforward tests for whether a graph is Hamiltonian.

We prove a few results here.

Theorem. If G is nontrivial and has a Hamilton path then $G \times K_2$ is Hamiltonian.

Proof. Let $P: v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ be a Hamilton path for G and let w_1, f, w_2 be the Hamiltonian (and Eulerian!) path for K_2. Recalling that the vertex set of $G \times K_2$ is $\{(v_i, w_j):\ 1 \le i \le n;\ 1 \le j \le 2\}$, we let $u_{ij} = (v_i, w_j)$. For the edges, e_{ij} joins vertices $u_{(i-1),j}$ and u_{ij} while f_i' joins u_{i1} and u_{i2}. Then

$$C:\ u_{11}, e_{21}, u_{21}, e_{31}, u_{31}, \dots, e_{n1}, u_{n1}, f_n', u_{n2}, e_{n2}, \dots, u_{32}, e_{32}, u_{22}, e_{22}, u_{12}, f_1', u_{11}$$

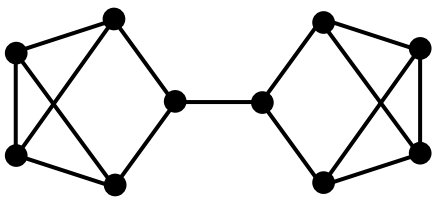
is a Hamilton cycle for $G \times K_2$.

Essentially, one forms the Hamilton cycle by traversing the Hamilton path in the first copy of G, using a K_2 edge, then traversing the Hamilton path in reverse in the second copy of G, and then using a K_2 edge to return to the starting vertex.

Corollary. For $n \ge 2$, the cube graphs Q_n are Hamiltonian.

Proof. We proceed by mathematical induction on n. For the base case, we observe that $Q_2 \cong C_4$ and so Q_2 is evidently Hamiltonian. Now, assume the result holds for $n = k$, i.e., assume Q_k is Hamiltonian. Then Q_k has a Hamilton path and so by the theorem, $Q_{k+1} \cong Q_k \times K_2$ is Hamiltonian.

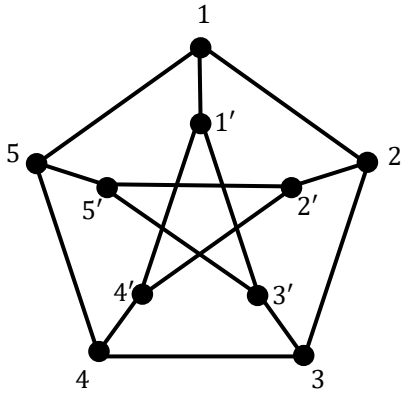
Not every cubic (3-regular) graph is Hamiltonian. Consider the easy example below:



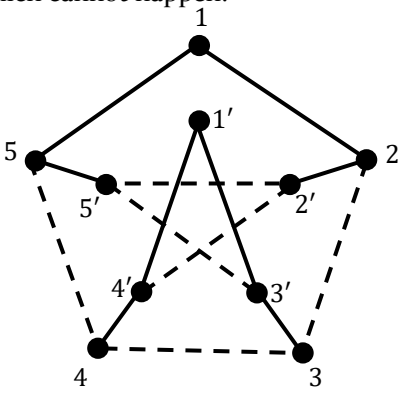
Observation. A graph with a bridge $e = uv$ is not Hamiltonian. By the results in II.B.3, every u, v-walk must use e. Hence, e would have to be on a supposed Hamiltonian cycle. But bridges cannot be on cycles, so no Hamiltonian cycle exists.

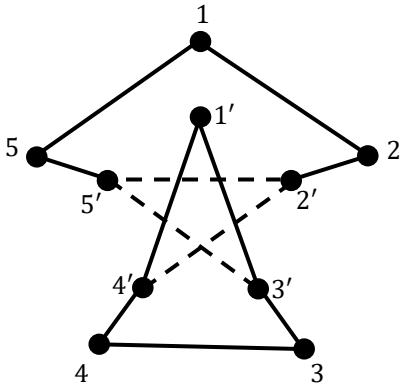
Another less-obvious example:

Proposition. The Petersen graph G is not Hamiltonian.



Sketch of proof. Suppose otherwise. Then any Hamilton cycle C would have to include either two or four radial edges (edges of the form aa'). If there are four, then WLOG assume edge $11'$ is not present in C. Then edges 12 and 15 must be present, along with edges $1'3'$ and $1'4'$. But then edges 23 and 45 must be absent, compelling edge 34 to be present, which means the 5-cycle with vertices $1', 3', 3, 4, 4'$ is a subgraph of C, which cannot happen.





If only two of the radial edges are present, there are two possibilities up to rotation: Either $22'$ and $55'$ are present, or $33'$ and $44'$ are present. If $22'$ and $55'$ are present, then every edge of the outer pentagon incident with $1, 3, 4$ must be present. But this forces every edge of the outer pentagon to be present and so $12, 23$, and $22'$ are all present, implying 2 is incident with three edges of the cycle, which is impossible. Similarly, if $33'$ and $44'$ are present, this forces all edges of the inner star incident with $1'$, $2'$, and $5'$ to be present. But this means $1'4', 2'4'$, and $44'$ are all present and so $4'$ is incident with three edges of the cycle, which is impossible.

□

II.D. GRAPH COLORING

II.D.1. VERTEX COLORING

There is a rich history of problems in graph theory that reduce to assigning colors to the vertices where no two adjacent vertices are assigned the same color. This is an extension of graphs being bipartite, where we can assign one color to all vertices in one part and a second color to all vertices in the second part.

We formalize the notion of a vertex coloring:

Definition. Given a graph G, a function $c: V \to \{1,2,3,\dots,k\}$ is said to be a k-**vertex coloring of** G if whenever u and v are adjacent, $c(u) \neq c(v)$.

Since the literature on vertex colorings is more extensive than that for edge colorings, the word "vertex" is often omitted, and so a k-**coloring** is actually a k-vertex coloring.

Definition. A graph G is k-**colorable** if there exists a k-coloring of G.

Definition. The quantity $\min\{k: G \text{ is } k\text{-colorable}\}$ is called the **chromatic number** of G. This is denoted $\chi(G)$. Recall that χ is the Greek letter "chi" and is the first letter in the word "chromatic" using Greek letters.

Observations.

- Conventionally, $\chi(N) = 0$ if N is a null graph.
- If G has loops, then no k-coloring exists for any k. Conventionally, $\chi(G) = \infty$ if G has a loop.
- A graph is k-colorable if and only if it is k-partite.
- $\chi(G) = 1$ iff G is totally disconnected and not null.
- For any nonempty bipartite graph G, $\chi(G) = 2$.
- For any $k \geq 1$, $\chi(C_{2k+1}) = 3$.
- If G has K_n as a subgraph, then $\chi(G) \geq n$.

Theorem II.D.1.a. For any loopless graph G, $\chi(G) \leq \Delta(G) + 1$.

Proof. Let $p = \Delta(G) + 1$ and $S = \{1,2,3,\dots,p\}$, the set of p colors. Consider the following algorithm:

1. Start with any assignment of colors from S to the vertices.
2. Assign $s :=$ the number of vertices adjacent to other vertices with the same color.
3. If $s = 0$, STOP; the assignment of colors is a p-coloring of G.
4. If $s > 0$, find a vertex v that is adjacent to another vertex with the same color.
5. Let S' be the set of colors used for vertices adjacent to v; we note $|S| < p$ because there are fewer than p vertices adjacent to v.
6. Reassign to v a color in $S - S'$. The vertex v is no longer adjacent to vertices with the same color.
7. Go to step 2.

Notice that when we return to step 2, the value of s must decrease. Since it cannot decrease below 0, this algorithm must stop.

▫

This bound is best possible as $\chi(C_{2k+1}) = 3 = \Delta(C_{2k+1}) + 1$.

The converse of the observation "if G has K_n as a subgraph, then $\chi(G) \geq n$" is not true.

Proposition. The Grötzsch graph G depicted in Figure II.D.1.b below satisfies $\chi(G) = 4$.

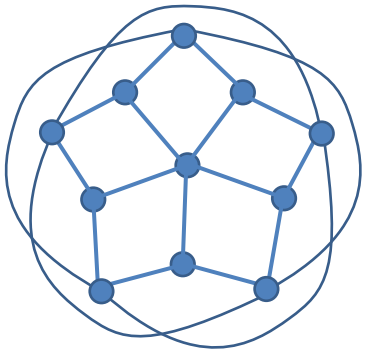


Figure II.D.1.b. The Grötzsch graph

Proof. Suppose we could 3-color the Grötzsch graph. Then the outermost vertices (forming a C_5) require all three colors. WLOG we use color 1 (red) for the topmost vertex and alternate colors 2 (black) and 3 (green) for the remaining vertices; see the left-most graph of Figure II.D.1.c. The blue vertices remain unassigned.

This then forces colors on some of the vertices at the next layer (center graph of Figure II.D.1.c.) Finally, the center vertex is adjacent to vertices that have been assigned each of colors 1, 2, and 3, necessitating a fourth color (right graph of Figure II.D.1.c.)

▫

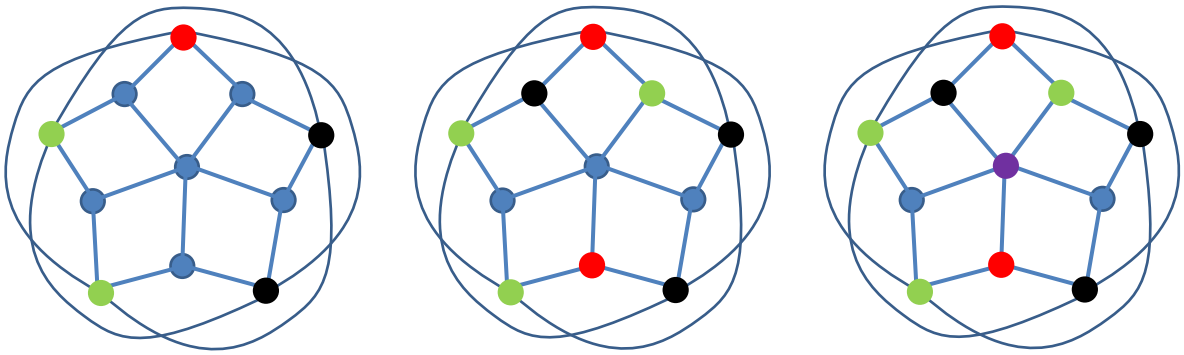


Figure II.D.1.c. Coloring the vertices of the Grötzsch graph.

Definition. A graph G is **triangle-free** if K_3 is not a subgraph of G.

Theorem II.D.1.d. For every $k \geq 1$, there exist triangle-free graphs such that $\chi(G) \geq k$.

Proof. We proceed by induction on k noting that C_5 is triangle-free with $\chi(G) \geq 1,2,3$ to establish the base case.

Now, suppose G is triangle-free with $\chi(G) = k$. Let $V(G) = \{v_1, v_2, \dots, v_n\}$. We construct a graph G' as follows:

1) Let $V(G') = \{v_1, v_2, \dots, v_n, w_1, w_2, \dots, w_n, x\}$
2) Let $E(G') = E(G) \cup \{v_i w_j : v_i v_j \in E(G)\} \cup \{w_i x : 1 \leq i \leq n\}$. In words, every edge of G is an edge of G'. Also, for each i with $1 \leq i \leq n$, we join v_i to w_j iff v_i is adjacent to v_j. Finally, all w_i vertices are joined to x.

Suppose G' has a triangle $u_0, e_1, u_1, e_2, u_2, e_3, u_0$. If any of the vertices $u_i = x$, then its neighbors are both w vertices, but the w vertices are not adjacent to each other. Similarly, if two of the vertices are w vertices, they would be adjacent, which cannot happen. If one of the vertices is w_i, then the other two vertices are neighbors v_{j_1} and v_{j_2} of v_i in G. But v_{j_1} and v_{j_2} are not adjacent because G is triangle-free. Therefore, w_i, v_{j_1} and v_{j_2} cannot be in a triangle. Thus, G' has no triangle.

Now, suppose we could n-color the vertices of G' with no two adjacent vertices the same color. This would mean at most $n - 1$ colors are used to color the w_i vertices, since x is adjacent to all of them. We can re-color v_i to be the same color as w_i as v_i has the same neighbors as w_i, except for the x vertex. But this means G is $(n-1)$-colorable, which contradicts $\chi(G) = n$. Thus, $\chi(G') \geq n + 1$.

Finally, if we n-color the vertices of G, color w_i the same as v_i and then color x with the $(n+1)^{st}$ color, we obtain an $(n+1)$-coloring of G'. Hence, $\chi(G') = n + 1$.

□

II.D.2. CHROMATIC POLYNOMIALS

Definition. Given a graph G, we define a function $p_G: \mathbb{Z} \to \mathbb{Z}$ called the **chromatic polynomial** of G. The value of $p_G(k)$ is the number of ways to k-color the vertices of G using colors from $\{1,2,3, \dots, k\}$.

Convention. If G is a null graph, then $p_G(k) = 1$, a constant function.

Example. Suppose G is a graph with n vertices $v_1, v_2, \dots, v_n$, all of which are isolated. Then we can color the vertices using the following list of subtasks:

- Color v_1 with any of the k colors.
- Color v_2 with any of the k colors.
- ⋮
- Color v_n with any of the k colors.

These n subtasks are all independent, so the number of ways to complete the list is the product of the numbers of ways to complete the individual subtasks. Hence, $p_G(k) = k^n$ for this graph.

Example. Suppose $G = P_n$, a path graph with n vertices (list them $v_1, \dots, v_n$.) Now, we can color the vertices using the following list of subtasks:

- Color v_1 with any of the k colors.
- Color v_2 with any of the $k-1$ colors not used to color v_1.
- Color v_3 with any of the $k-1$ colors not used to color v_2.
- ⋮
- Color v_n with any of the $k-1$ colors not used to color v_{n-1}.

Again, these n subtasks are all independent, so the number of ways to complete the list is the product of the numbers of ways to complete the individual subtasks. Hence, $p_G(k) = k(k-1)^{n-1}$.

Example. Suppose $G = K_n$, the complete graph on n vertices, $v_1, v_2, \dots, v_n$. We can color the vertices using the following list of subtasks:

- Color v_1 with any of the k colors.
- Color v_2 with any of the $k-1$ colors not used to color v_1.
- Color v_3 with any of the $k-2$ colors not used to color v_1 or v_2.
- Color v_4 with any of the $k-3$ colors not used to color v_1, v_2, or v_3.
- ⋮
- Color v_n with any of the $k-(n-1)$ colors not used to color the other vertices.

Hence, for $k \geq n$, $p_G(k) = k(k-1)(k-2)\cdots\big(k-(n-1)\big) = \frac{k!}{(k-n)!}$. For $k < n$, $p_G(k) = 0$.

Observation. Given graphs G and H, recall that $G \sqcup H$ is their disjoint union. We then have

$$p_{G \sqcup H}(k) = p_G(k) \cdot p_H(k).$$

Observation. The chromatic polynomial of a graph is the product of the chromatic polynomials of its components.

Observation. If G has loops, then $p_G(k) = 0$, a constant function.

Observation. If G has parallel edges and H is the underlying simple graph of G, then $p_G(k) = p_H(k)$.

Theorem II.D.2.a. Let G be a simple graph and $e = uv$ be an edge of G. Let $H_1 = G - e$ and let $H_2 = G/e$ be the graph obtained by contracting the edge e (recall this is achieved by identifying its endpoints u and v.) Then $p_G(k) = p_{H_1}(k) - p_{H_2}(k)$.

Proof. Consider a k-coloring of H_1. If u and v are colored using different colors, then we can insert e, producing a k-coloring of G. This process is reversible, so $p_G(k)$ is the number of k-colorings of H_1 where u and v are colored with different colors. If u and v are colored using the same color, then we can contract edge e, identifying vertices u and v, and producing a k-coloring of H_2. This process is also reversible and so the number of k-colorings of H_1 with u and v colored the same is $p_{H_2}(k)$. It follows that $p_{H_1}(k) = p_G(k) + p_{H_2}(k)$ which implies the result.

□

Corollary II.D.2.b. For any $n \geq 4$,

$$\begin{aligned} p_{C_n}(k) &= p_{P_{n-1}}(k) - p_{C_{n-1}}(k) \\ &= k(k-1)^{n-1} - p_{C_{n-1}}(k). \end{aligned}$$

Theorem II.D.2.c. For any simple graph G with n vertices, the function $p_G(k)$ is a monic (leading coefficient = 1) polynomial of degree n in k.

Proof. We proceed by mathematical induction on the number of edges m in G. If $m = 0$, then we have $p_G(k) = k^n$, a monic polynomial of degree n in k. If $m > 0$, then we assume the result holds for all graphs with $m - 1$ edges. If G is a graph with m edges, then let $e = uv$ be an edge. We then have

$$p_G(k) = p_{H_1}(k) - p_{H_2}(k)$$

where $H_1 = G - e$ and $H_2 = G/e$. These graphs both have $m - 1$ edges, so the result holds for them. Furthermore, $p_{H_1}(k)$ is a monic degree n polynomial while $p_{H_2}(k)$ is a monic degree $n - 1$ polynomial. Their difference is a monic degree n polynomial and the result follows.

□

Observation. The graph G is r-colorable if and only if $p_G(r) \neq 0$.

Observation. The graph G is not r-colorable if and only if $k - r$ is a factor of $p_G(k)$.

II.D.3. EDGE COLORING

We now consider coloring the edges of the graph, now with the restriction that any two edges incident with the same vertex are colored with different colors.

We formalize the notion of an edge coloring:

Definition. Given a graph G, a function $c: E \to \{1,2,3,\dots,k\}$ is said to be a k-**edge coloring of G** if whenever e and f have a common endpoint, $c(e) \neq c(f)$.

Definition. A graph G is k-edge **colorable** if there exists a k-edge coloring of G.

Definition. The quantity $\min\{k: G \text{ is } k\text{-edge colorable}\}$ is called the **edge chromatic number** or **chromatic index** of G. This is denoted $\chi'(G)$.

Theorem II.D.3.a (Vizing). For any finite loopless graph G, $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.

Proof. Since the edges incident with any vertex of degree $\Delta(G)$ must all be assigned different colors, the lower bound is immediate.

For the upper bound, we assume counterexamples exist and let G be a counterexample with the minimum number of edges. This means for any edge e, $\chi'(G-e) \leq \Delta(G-e)+1 \leq \Delta(G)+1$. If for some edge e, we have $\chi'(G-e) \leq \Delta(G)$, then we could $\Delta(G)$-edge color $G-e$, assign a new color to e, and obtain a $(\Delta(G)+1)$-edge coloring of G as a result, which cannot happen. Thus, for any edge e, $\chi'(G-e) = \Delta(G)+1$.

Let $e = vw_0$ and let $G-e$ be $(\Delta(G)+1)$-edge colored, using colors $\{1,2,3,\dots,\Delta(G)+1\}$. Our goal is to color vw_0 and adjust the colors on other edges to construct a $(\Delta(G)+1)$-edge coloring of G.

We construct finite sequences of edges and colors:

$$vw_0, vw_1, vw_2, \dots, vw_k; \quad c_0, c_1, c_2, \dots, c_k$$

where c_i is a color not used for any of the edges incident with w_i and vw_{i+1} is colored with c_i. Note that since every vertex has degree at most $\Delta(G)$, c_i exists; thus, the sequences can be built. We stop the sequence when either c_k is not used for any edge incident with v or when $c_k = c_j$ for some $j, 0 \leq j \leq k-1$.

If c_k is not used at v, then we can color vw_i with $c_i, 0 \leq i \leq k$, and produce a $(\Delta G+1)$-edge coloring of G.

If $c_k = c_j$ for some $j, 0 \leq j \leq k-1$, then color vw_i with $c_i, 0 \leq i \leq j-1$ and let c_ℓ be an unused color at v. We note that c_k is not used at w_j or w_k. We must reassign a color to vw_j. There are some easy cases to consider:

1) If c_k is not used at v, then color vw_j with c_k.
2) If c_ℓ is not used at w_j, then color vw_j with c_ℓ.
3) If c_ℓ is not used at w_k, then color vw_i with c_i for $j \leq i \leq k-1$ and color vw_k with c_ℓ; note that none of the $vw_i, j \leq i \leq k-1$ are colored with either c_k or c_ℓ.

If these all fail, then we look at the subgraph G' of G induced by the edges colored c_k or c_ℓ. Every vertex has degree at most 2 and so G' has components that are either paths or cycles. Look at the vertices v, w_j, and

w_k. Since c_ℓ is not used at v, v must be at the end of a path whose initial edge is colored c_k in G'. Since c_k is not used at either w_j or w_k, these vertices must be at the end of paths whose initial edges are colored c_ℓ in G'. Thus, not all three vertices can be in the same component of G', implying at least one is alone in its component. Suppose v is alone. Then switching the roles of c_k and c_ℓ in that component will cause case 2) above to hold. If w_j is alone, switching the roles of c_k and c_ℓ in that component will cause case 2) above to hold. Finally, if w_k is alone, switching the roles of c_k and c_ℓ in that component will cause case 3) above to hold.

Thus, G has a $(\Delta G + 1)$-edge coloring, which is a contradiction and so counterexamples do not exist.

□

Definition. Graphs such that $\chi'(G) = \Delta(G)$ are called **type-1 graphs** while those such that $\chi'(G) = \Delta(G) + 1$ are called **type-2 graphs.**

Definition. A bridgeless cubic type-2 graph is called a **snark.** (Named after the Lewis Carroll poem, "The Hunting of the Snark."

Proposition II.D.3.b. The Petersen graph is a snark.

Proof. In Figure II.D.3.c, we show the Petersen graph G.

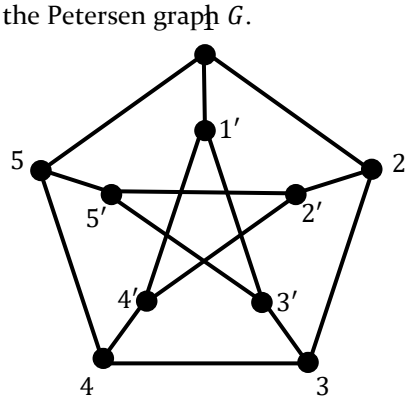


Figure II.D.2.c. The Petersen Graph

Suppose $\chi'(G) = 3$. Suppose at least six edges are colored the same color. If they did not share any endpoints, then there would be at least twelve distinct vertices. Since G has only ten vertices, this cannot happen. Hence, no color class has more than five edges. Since G has 15 edges and there are three color classes, each color class must have exactly five edges.

WLOG, assume the outer pentagon is colored with edge 34 red, edges 45 and 12 blue, and edges 15 and 23 green (left graph of Figure II.D.2.d). This forces edges $11'$, $22'$, and $55'$ to be red (right graph of Figure II.D.2.d). But every edge of the center pentagon is incident with one of those three red radial edges, implying no edge in that pentagon is red. Thus, the red color class has only four edges which is a contradiction.

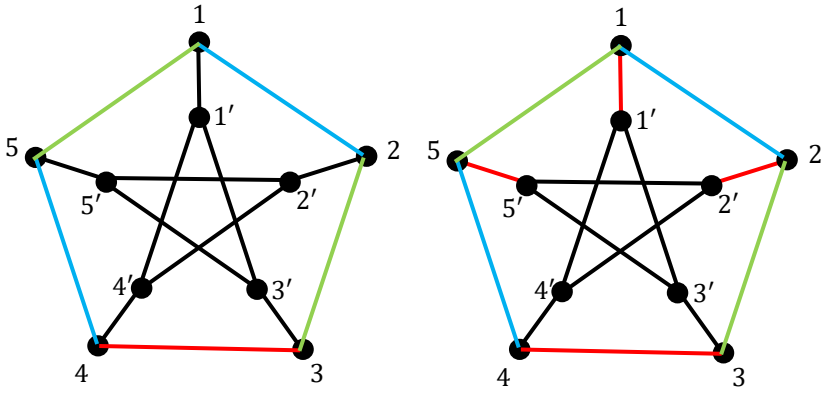


Figure II.D.3.d. Attempting to 3-Edge Color the Petersen Graph

It follows that $\chi'(G) > 3$. By Vizing's theorem, $\chi'(G) = 4$.

□

SECTION III. TREES AND FORESTS

III.A. PRELIMINARIES

III.A.1. DEFINITIONS

Definition. A **forest** is a graph that has no cycles (including loops or pairs of parallel edges) as subgraphs. Such a graph is also called **acyclic**.

Definition. A **tree** is a connected acyclic graph. See Figure III.A.1.a for examples of trees.

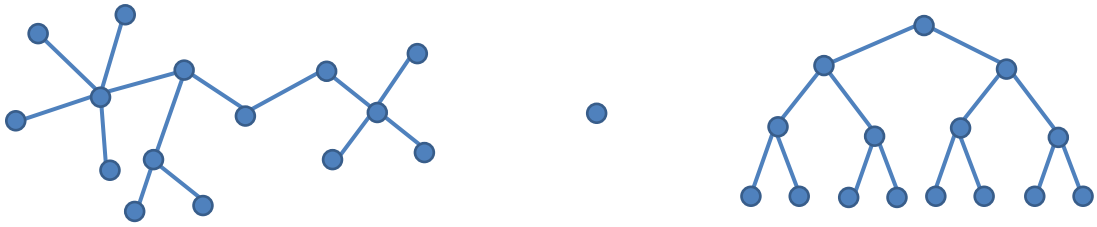


Figure III.A.1.a. Three examples of trees.

Observations.

- If G is a forest, then each component of G is a tree.
- A trivial (and empty) graph is a tree.
- A path is a tree.
- The only complete graphs that are trees are K_1 and K_2.
- If the complete bipartite graph $K_{n,m}$ is a tree, then $n = 1$ or $m = 1$.
- Forests are bipartite; this follows from the fact that forests have no odd cycles because they have no cycles at all.

III.A.2. PATH UNIQUENESS

A useful property of trees is the following:

Theorem III.A.2.a. Given a tree T, for any two vertices u and v, there is a unique u,v-path.

Proof. Existence is immediate, since T is connected by definition. For uniqueness, we let P and Q be two u,v-paths:

$$P\colon\ u = p_0, e_1, p_1, \dots, p_{k-1}, e_k, p_k = v$$
$$Q\colon\ u = q_0, f_1, q_1, \dots, q_{\ell-1}, f_\ell, q_\ell = v$$

We want to show that $P = Q$.

Since T has no cycles, every edge e_i, f_j is a bridge, implying every u,v-path must contain every edge e_i and every edge f_j. Hence, $\{e_1, e_2, \dots, e_k\} = \{f_1, f_2, \dots, f_\ell\}$ as sets. Since the e_i are distinct and the f_j are distinct,

and these sets must have the same cardinality, we have $k = \ell$. Also, since the edge sets are equal, the sets of their endpoints must also be equal, and so $\{p_0, p_1, \dots, p_k\} = \{q_0, q_1, \dots, q_k\}$.

We claim for all i, $0 \le i \le k$, $p_i = q_i$. First, we have that $p_0 = q_0 = u$. We suppose the claim fails and we let j be the smallest counterexample, i.e., $p_j \ne q_j$ and for all $i, 0 \le i \le j-1$, $p_i = q_i$. We note that for $1 \le i \le j-1$, $e_i = f_i$. Let $q_j = p_h$. We note that $e_j = p_{(j-1)}p_j \ne q_{j-1}q_j = f_j$, implying $j \ne h$ and e_j is distinct from $e_1 = f_1, e_2 = f_2, \dots, e_{j-1} = f_{j-1}$, and so $j \le h$. Hence, $j < h$. If we delete e_j, then u and v become disconnected. On the other hand, the walk

$$R: u = q_0, f_1, q_1, \dots, q_{j-1}, f_j, q_j = p_h, e_{h+1}, p_{h+1}, \dots, e_k, p_k = v$$

is a u, v-walk, which contradicts u and v being disconnected. Thus, the claim that $p_i = q_i$ for all $i, 0 \le i \le k$ is true and so $P = Q$.

□

Corollary III.A.2.b. If F is a forest, then for any two vertices u and v, there exists at most one u, v-path.

Corollary III.A.2.c. Given two vertices u and v of a tree T, there is a unique u, v-walk of length $D(u, v)$, namely, the unique u, v-path guaranteed by Theorem III.A.2.a.

Corollary III.A.2.d. Given three vertices u, v, w of a tree T, if $D(u, v) = D(u, w) + D(w, v)$, then w is a vertex on the unique u, v-path.

Proof. Let W_1 be the unique u, w-walk of length $D(u, w)$ and W_2 be the unique w, v-walk of length $D(w, v)$. Then the concatenation W_1W_2 is a u, v-walk W of length $D(u, v)$. By corollary III.A.2.c, W is the only walk of length $D(u, v)$ which must necessarily be the unique u, v-path.

III.A.3. THE EDGE FORMULA

It turns out that the order, size, and number of components of a forest always obey a simple relationship:

Theorem III.A.3.a. If F is a forest with n vertices, m edges, and k components, then $n = m + k$.

Proof. Let F be as hypothesized. This is a graph with exactly k components. Since F has no cycles, every edge of F is a bridge by Theorem II.B.3.h. If we delete the m edges one at a time, the resulting graph $F - E$ will have $m + k$ components by Theorem II.B.3.g. But $F - E$ is a completely disconnected graph on n vertices, so $F - E$ has n components. It follows that

$$n = m + k$$

which is the desired result.

□

This theorem leads to an often-used fact about trees:

Corollary III.A.3.b. The order of a tree is one greater than its size.

A converse of Theorem III.A.2.a also holds:

Theorem III.A.3.c. If G has n vertices, m edges, and k components and $n = m + k$, then G is a forest.

Proof. Delete non-bridges from components of G one at a time until all remaining edges are bridges; we let $e_1, e_2, \dots, e_j$ be the resulting sequence of non-bridges that are deleted. Then $G - \{e_1, \dots, e_j\}$ is a forest with n vertices, $m - j$ edges, and k components. From Theorem III.A.3.a,

$$n = (m - j) + k.$$

But since $n = m + k$, we conclude that $j = 0$ and so G has no non-bridges, and the result follows.

□

Corollary III.A.3.d. If G is connected and has order one higher than its size, then G is a tree.

Summarizing the corollaries, we have:

Theorem III.A.3.e. Let G be a graph with n vertices and m edges. Any two of the following together imply the remaining statement:

- G is acyclic
- G is connected
- $n = m + 1$

and G is a tree iff all three hold.

III.A.4. LEAVES

Definition. A **leaf** is a vertex of degree 1.

Observation. If T is a trivial tree, then its vertex is not a leaf.

The following fundamental result can be proved in various ways. The proofs offered here demonstrate a variety of proof techniques that are commonly used in graph theory. The first proof uses observations about specific degrees of vertices and the handshaking lemma. The second proof uses a maximality argument. The third proof uses strong mathematical induction. The last proof derives this result as a corollary of a result from a previous section and a "smallest counterexample" argument.

Theorem III.A.4.a. Every nontrivial tree T has at least two leaves.

First Proof. Let T be a tree with $n \geq 2$ vertices. By the previous results, T has $n - 1$ edges. For $k \geq 0$, let n_k be the number of vertices of degree k. We observe that

$$\sum_{k\geq 0} n_k = n$$

as the LHS simply looks at the numbers of vertices of each degree; adding these values yields the number of vertices. Also, the total degree is reflected in the following equation:

$$\sum_{v\in V} \deg v = \sum_{k\geq 0} kn_k$$

and from the Handshaking Lemma,

$$2(n-1) = 2m = \sum_{v\in V} \deg v.$$

Combining these results, we have

$$\sum_{k\geq 0} kn_k = -2 + \sum_{k\geq 0} 2n_k$$

or

$$2 = \sum_{k\geq 0} (2-k)n_k = 2n_0 + n_1 + \sum_{k\geq 2} (2-k)n_k.$$

We note that $n_0 = 0$ because there can be no isolated vertices. Therefore,

$$n_1 = 2 + \sum_{k\geq 2} (k-2)n_k \geq 2$$

since each term in the sum is nonnegative.

□

Second Proof. Let $P: v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_k$ be a longest path in T. We note that since T is nontrivial, we must have $k \geq 1$ and hence, $v_0 \neq v_k$.

Suppose v_0 is not a leaf. Then $\deg(v_0) \geq 2$ and so there is some vertex $v_{-1} \neq v_1$ that is adjacent to v_0 via an edge e_0. If $v_{-1} = v_j$ is on P, then

$$C\colon\ v_{-1}, e_0, v_0, \ldots, v_{j-1}, e_j, v_j$$

is a cycle in T which contradicts T being acyclic. If v_{-1} is not on P, then

$$Q\colon v_{-1}, e_0, v_0, \ldots, v_{k-1}, e_k, v_k$$

is a longer path than P in T which contradicts P being a longest path. Therefore, v_0 must be a leaf.

We can apply the same reasoning to show that v_k must be a leaf.

Since v_0 and v_k are distinct leaves, there must be at least two leaves.

▫

Third Proof. We proceed by strong mathematical induction on the number of vertices $n \geq 2$. A tree with 2 vertices must be isomorphic to K_2; each vertex in K_2 has degree 1 and so both vertices are leaves, establishing the base case.

For the inductive step, we assume the result holds for all k with $2 \leq k \leq n-1$ and let T be a tree with n vertices. Any edge $e = uv$ is a bridge and so the graph $T - e$ must have two components T_1 and T_2, each one a tree. We assume u is a vertex of T_1 and v is a vertex of T_2. There are three cases to consider, depending on how many of these components are trivial trees.

Case 1. Both T_1 and T_2 are trivial trees. Then T is isomorphic to K_2 which is the base case.

Case 2. One of the components is trivial; WLOG, we assume T_1 is trivial and T_2 is nontrivial. Then u is an isolated vertex of $T - e$, implying $\deg(u) = 1$ in T and so u is a leaf. By induction, there must be two leaves in T_2. One of these leaves, w say, must be distinct from v and so w is a second leaf of T.

Case 3. Neither component is trivial. Then by induction, each of the components has two leaves. Let $x \neq u$ be a leaf of T_1 and $w \neq v$ be a leaf of T_2. Then x and w are leaves of T.

▫

Fourth Proof. Assume there are counterexamples and let T be a counterexample with the smallest number of edges. If T has no leaves, then every vertex is at least degree 2. From Lemma II.C.1.d, T has a cycle which is contradictory. Hence, T has at least one leaf ℓ. If we delete the edge e incident with ℓ, then $T - e$ has two components, one of them consisting solely of ℓ. Hence, $T - \ell$ is a tree. But then either $T - \ell$ is trivial, in which case T is a K_2 graph and both vertices are leaves, or $T - \ell$ is nontrivial. In this latter case, $T - \ell$ has fewer edges than T and so it has at least two leaves x and y. Restoring ℓ will leave at least one of x or y intact as a leaf and so T has at least two leaves, ℓ and one of x or y. Hence, T is not a counterexample which is a contradiction. Hence, there are no counterexamples.

▫

Definition. If $H \leq G$ and $V(H) = V(G)$, then H is called a **spanning subgraph** of G. The subgraph H is said to **span** G.

This concept is used most often in practice when H is a tree; a **spanning tree** is a tree that is a subgraph of G that also spans G.

Theorem III.A.5.a. Every connected graph G with finitely many edges has spanning trees.

Proof. Let G be connected. We define a sequence of connected graphs

$$G_0, G_1, \ldots$$

recursively, setting $G_0 = G$. We will be deleting edges only from G, so the vertex sets of G_i are all the same. For a given index i, if G_i has a non-bridge e, then define $G_{i+1} = G_i - e$. If every edge of G_i is a bridge, then G_i is a tree and, since $V(G_i) = V(G)$, G_i is a spanning tree of G. Finally, since G has finitely many edges, this sequence must terminate at or before G_m where m is the number of edges in G.

▫

Theorem III.A.5.b. Let G be a connected graph with finitely many edges. Every bridge of G lies on every spanning tree of G.

Proof. Let H be a spanning tree and $e = uv$ be a bridge of G. Since H is a spanning tree, u and v are vertices of H and so there is a unique u, v-path P in H. But P is also a u, v-path in G which must contain e, since e is a bridge of G. Hence, e is an edge of H.

▫

A commonly used operation is adding an edge to a spanning tree.

Theorem III.A.5.c. Let G be a connected graph and T be a spanning tree of G. For any edge e of G that is not one of T, the graph $T + e$ has a unique subgraph that is a cycle.

Proof. Suppose G has n vertices. Then T has $n - 1$ edges and so $T + e$ would have n edges. It follows that $T + e$ is not a tree and since $T + e$ is connected, $T + e$ is not acyclic. This establishes existence of a cyclic subgraph C.

Suppose $T + e$ has another cyclic subgraph D distinct from C. The edges of D cannot all belong to C; C has only one cyclic subgraph, namely itself. So, there is some edge f of D that is not on C. Since f is on a cycle, it is not a bridge and so $H = T + e - f$ is connected. This new graph H has n vertices, $n - 1$ edges and is connected and so it is a tree. But it also has C as a subgraph, which contradicts H being acyclic. Hence, D does not exist and C is unique.

▫

III.B. ROOTED TREES

Rooted trees are used extensively in computer science applications.

III.B.1. DEFINITIONS AND PRELIMINARY RESULTS

Definition. A **rooted tree** (T, r) is a tree T with a distinguished vertex r called the **root**.

Convention. Given a rooted tree, it is conventional to denote the root with the variable r.

Some commonly used terms to describe relationships among vertices in rooted trees are borrowed from genealogy.

Definitions. Given a rooted tree (T, r) and two distinct vertices u and w,

- If u is a vertex on the unique r, w-path, then u is an **ancestor** of w and w is a **descendant** of u.
- If u and w are adjacent as well, then u is the **parent** of w and w is a **child** of u.
- If u and w have the same parent, then they are **siblings**.
- If u is the parent of w's parent, then u is the **grandparent** of w and w is a **grandchild** of u.
- (Less common). If u and w have the same grandparent but not the same parent, then they are **cousins**.
- (Less common). If u and the parent of w are siblings, then u is the **uncle (or aunt)** of w and w is the **nephew (or niece)** of u.

Observations.

- The root has no ancestors.
- The root is an ancestor of every non-root.
- Every non-root is a descendant of the root.
- The only non-child is the root.
- If T is nontrivial, then every non-parent is a leaf.
- It is possible for the root to have only one child, making it a vertex of degree 1, and so it would be a leaf by the established definition. Hence, "non-parent" is more precise than "leaf" in general in this context. Nonetheless, "leaf" is used more commonly.

Lemma III.B.1.a. For every edge e of a rooted tree (T, r) one endpoint is the parent of the other.

Proof. Let $e = uv$. If u is on the unique r, v-path, then the path must end $\ldots, u, e, v$ and so u is the parent of v. Similarly, if v is on the unique r, u-path, then v is the parent of u.

If neither happens, then we can trace a u, v-walk (not necessarily a path) without e by traversing the unique u, r-path followed by the unique r, v-path. Deleting e must disconnect its endpoints since e is a bridge, but this does not happen, a contradiction. Thus, this case cannot happen and the result follows.

□

Definition. Given a rooted tree (T, r), let child: $E \to V - \{r\}$ be the function where child(e) is the endpoint of e that is the child of the other endpoint of e.

Lemma III.B.1.b. Given a rooted tree (T, r), the function child: $E \to V - \{r\}$ is a bijection.

Proof. First, this function is surjective since if v is a child with parent u, then child$(uv) = v$. No child is left behind. Next, suppose child$(e) = v =$ child(f). Since there is a unique r, v-path, the last edge in this path must be both e and f, simultaneously, implying $e = f$. This establishes injectivity. ▫

This formalizes the observation that every child can be associated with the edge joining it to its parent.

Corollary III.B.1.c. For a rooted tree, the number of children equals the number of edges.

This is immediate from Corollary III.A.3.b as well as from Lemma III.B.1.b, because the number of edges is always 1 lower than the number of vertices in a tree. Since the root is the only non-child, the number of children is also 1 lower than the number of vertices in a rooted tree. ▫

III.B.2. LEVEL, HEIGHT, AND SUBTREES

Definition. Let (T, r) be a rooted tree. The **level** of a vertex v is $D(r, v)$, the distance from the root to v; recall that $D(r, v)$ is the length of a shortest r, v-path. Since there is only one r, v-path, the distance must be the length of this path.

Observations.

- The root is a level 0 vertex.
- For any level L vertex v, any of its children are level $L + 1$ vertices and its parent (if it has one) is a level $L - 1$ vertex.
- If u is a level L vertex and w is a level L' vertex, then $D(u, w) \leq L + L'$.

Definition. The **height** of a rooted tree (T, r) is the maximum level among its vertices.

Observations. Suppose a rooted tree (T, r) has height H.

- Every level H vertex is a non-parent.
- For any two vertices u and w, $D(u, w) \leq 2H$.
- If T has n vertices and m edges, $H \leq n - 1 = m$.

There are many applications of rooted trees that make use of subtrees induced by a vertex and all its descendants. Invariably, these subtrees are treated as rooted trees, so we formalize this now.

Definition. Let (T, r) be a rooted tree and v be a given vertex of T. Also, let $S(v)$ be the set containing v and all descendants of v. Then the ***v*-subtree of (T, r)** is the rooted tree (T', v) where T' is the subgraph of T induced by $S(v)$.

Definition. If v is a child of r, then the v-subtree of (T, r) is said to be a **principal subtree**.

Figure III.B.2.a shows a rooted tree of height 3 and its principal subtrees.

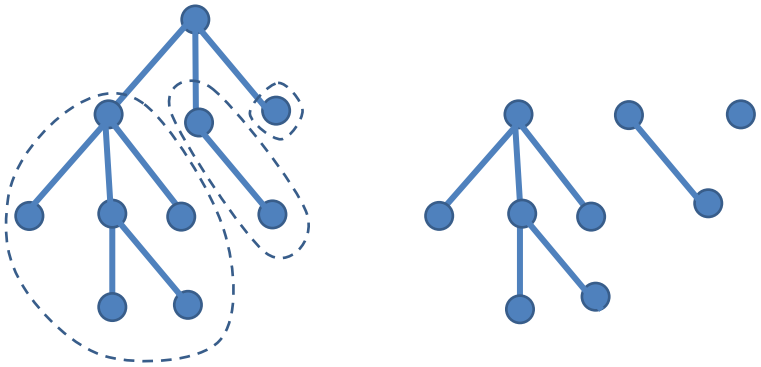


Figure III.B.2.a. A rooted tree of height 3 and its principal subtrees.

Definition. If T is a binary tree, then for any parent v with two children, we designate one of its children to be the **left child** and the other one to be the **right child**. If v has one child, we allow it to have either designation.

Definition. Given a binary tree T with root r, the subtree rooted at its left child is called the **left subtree** and the subtree rooted at its right child is called the **right subtree**.

In Figure III.B.2.b, we get an intuitive picture that distinguishes a "balanced" from an "unbalanced" rooted tree. A "balanced" rooted tree tends to be more efficient at storing and retrieving data.

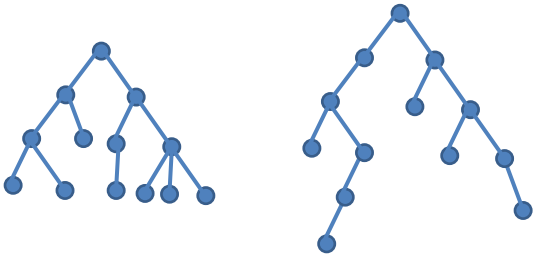


Figure III.B.2.b. Balanced vs. unbalanced rooted trees

Definition. A rooted tree (T, r) is said to be **height balanced** if for every vertex v—letting $\{w_1, w_2, \ldots, w_k\}$ be the set of children of v—the heights of the w_i-subtrees are within 1 of each other.

Definition. A rooted tree (T, r) of height H is said to be **strongly height balanced** if every non-parent is a level $H - 1$ vertex or a level H vertex.

III.B.3. BINARY AND n-ARY TREES

Applications of binary trees abound in computer science, especially concerning searching and sorting algorithms. Binary and n-ary trees are analogous to regular graphs—there are only two trees that are regular graphs, so there is little point in studying regular trees—and so there are many analogous results that involve such trees.

Definition. Let n be a non-negative integer. An ***n*-ary tree** is a rooted tree (T, r) such that every parent has at most n children.

Special Cases. A 0-ary tree is trivial. A **unary tree** is a 1-ary tree; such a tree is a path with the root at an endpoint. A **binary tree** is a 2-ary tree, and a **ternary tree** is a 3-ary tree.

Definition. A vertex of an n-ary tree is **saturated** if it has n children.

Definition. An n-ary tree is **full** if every parent is saturated.

In Figure III.B.3.a, both trees are ternary (3-ary) trees and the tree on the right is a full ternary tree.

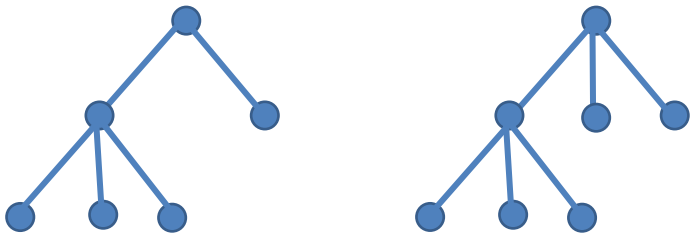


Figure III.B.3.a. Ternary Trees; the Right Tree is a Full Ternary Tree

Observations.

- A trivial rooted tree is an example of an n-ary tree for any non-negative integer n.
- If (T, r) is an n-ary tree, then it is also an $(n+1)$-ary tree.
- For a nontrivial full n-ary tree, the root has degree n, all other parents have degree $n+1$, and all non-parents have degree 1.
- If Δ is the maximum degree of a rooted tree (T, r), then (T, r) is a $(\Delta - 1)$-ary tree.

Theorem III.B.3.b. Given a full n-ary tree (T, r), if there are k parents, then there are $(n-1)k+1$ non-parents.

Proof. Let ℓ be the number of non-parents. If T is trivial, we have $k = 0$, $\ell = 1 = (n-1)0 + 1$, as desired. If T is nontrivial, then by the third observation above, the total degree of T is

$$n + (n+1)(k-1) + \ell = (n+1)k + \ell - 1$$

and the number of edges is $k + \ell - 1$. Hence, by the handshaking lemma,

$$2k + 2\ell - 2 = (n+1)k + \ell - 1.$$

Solving this for ℓ, we obtain

$$\ell = (n-1)k + 1,$$

as desired.

▫

Theorem III.B.3.c. For integers $n \geq 1$ and $L \geq 0$, an n-ary tree has at most n^L vertices at level L.

Proof. Let T be an n-ary tree and for each $L \geq 0$, let N_L be the number of vertices at level L. We proceed by induction on L. For the base case, there is $1 = n^0$ vertex at level 0, namely the root. Thus, $N_0 = 1$ is established.

Now, for the inductive step, assume $N_L \leq n^L$. Since every vertex at level $L+1$ is a child of a vertex at level L and that every vertex at level L has at most n children, we have $N_{L+1} \leq n\, N_L \leq n \cdot n^L = n^{L+1}$, completing the inductive step and the proof. ▫

Corollary III.B.3.d. For $n \geq 2$ and $H \geq 0$, if T is an n-ary tree of height H, then T has at most $\frac{n^{H+1}-1}{n-1}$ vertices.

Proof. Let N_L be the number of vertices at level L. Then

$$|V| = \sum_{L=0}^{H} N_L \leq \sum_{L=0}^{H} n^L = \frac{n^{H+1}-1}{n-1}$$

as desired. Furthermore, we have equality iff $N_L = n^L$ for all levels L. ▫

We can derive a lower bound for H by isolating H:

$$\begin{aligned} |V| &\leq \frac{n^{H+1}-1}{n-1} \\ (n-1)\,|V| &\leq n^{H+1}-1 \\ (n-1)\,|V|+1 &\leq n^{H+1} \\ \log_n\big((n-1)\,|V|+1\big) &\leq H+1 \\ \lceil \log_n\big((n-1)\,|V|+1\big) \rceil &\leq H+1. \end{aligned}$$

and so

$$H \geq \lceil \log_n\big((n-1)\,|V|+1\big) \rceil - 1.$$

For binary trees,

$$H \geq \lceil \log_2(|V|+1) \rceil - 1.$$

III.B.4. BINARY ALGORITHMS

Observation. If T is a rooted tree with ℓ leaves (we assume the root is not a leaf), then there are ℓ paths with the root and a leaf as endpoints.

Definition. A **binary algorithm** is a sequence of steps where at each step where a choice of how to proceed exists, called a **decision point**, there are two choices of how to proceed. An **exit point** is a step where the algorithm stops.

We can use binary trees to model binary algorithms as follows:

- The root is the first decision point.
- Every exit point is a leaf.
- Every decision point is a parent of the immediately following decision points or exit points.

Application. Storing data in a binary tree.

Suppose we have a list of n ($n \geq 2$) distinct items $\{a_0, a_1, a_2, \dots, a_{n-1}\}$ that can be ordered, i.e., for any two a_i, a_j, either $a_i < a_j$ or $a_j < a_i$. Then we can build a binary tree T to contain the list. This tree is unlikely to be balanced.

1. Place a_0 at the root.
2. Assign $i := 1$.
3. Assign $T' := T$
4. Assign $r' :=$ root T'; $a' =$ value assigned to r'
5. If $a_i < a'$
    a. If r' has a left child
        i. Assign $T' :=$ left_tree(T')
        ii. Go to 4.
    b. If r' has no left child
        i. Place a_i at left_child(r')
        ii. $i++$
        iii. If $i \leq n$ go to 3; else STOP.
6. If $a_i > a'$
    a. If r' has a right child
        i. Assign $T' :=$ right_tree(T')
        ii. Go to 4.
    b. If r' has no right child
        i. Place a_i at right_child(r')
        ii. $i++$
        iii. if $i \leq n$ go to 3; else STOP

Application. Bounds for binary searching algorithms.

Consider the problem of designing a binary algorithm to determine whether an item a is in a list of items $\{a_1, a_2, \dots, a_n\}$ already in a binary tree T. We could run the following pseudocode:

1. Assign $T' := T$, $r' := r$
2. Assign a' to the value at r'
3. If $a = a'$ REPORT "in list" and STOP
4. If $a < a'$
    a. If r' has no left child REPORT "not in list" and STOP
    b. If r' has a left child
        i. Assign $T' :=$ left_tree(T')
        ii. Assign $r' :=$ left_child(r')

iii. Go to 2

5. If $a > a'$
   a. If r' has no right child REPORT "not in list" and STOP
   b. If r' has a right child
      i. Assign $T' \coloneqq \text{right\_tree}(T')$
      ii. Assign $r' \coloneqq \text{right\_child}(r')$
      iii. Go to 2

The worst-case running time is the length of a longest r, v-path in T (plus 1, if the value is not on the tree and we have to trace a longest path to determine this.) This would be $H + 1$ where $H = \text{height}(T)$. From before, $H + 1 \geq \lceil \log_2(n+1) \rceil$. This gives us a lower bound of $\lceil \log_2(n+1) \rceil$ for worst-case behavior for detecting whether an item is one of a list of n items.

Application. Bounds for binary sorting algorithms.

Consider the problem of designing a binary algorithm to sort a list of n distinct items. Such a list can be presented in any of $n!$ orders and so a sorting algorithm must be capable of distinguishing $n!$ possible inputs. In this application, we envision making binary choices based on comparing elements in the list and, ultimately, reaching a leaf that discloses the order of the elements in the list. For instance, consider the following algorithm for $n = 3$, letting the list be $\{a_1, a_2, a_3\}$:

1. If $a_1 < a_2$, then
   a. If $a_2 < a_3$, then $a_1 < a_2 < a_3$
   b. If $a_3 < a_2$, then
      i. If $a_1 < a_3$, then $a_1 < a_3 < a_2$
      ii. Otherwise, $a_3 < a_1 < a_2$
2. If $a_2 < a_1$, then
   a. If $a_1 < a_3$, then $a_2 < a_1 < a_3$
   b. If $a_3 < a_1$, then
      i. If $a_2 < a_3$, then $a_2 < a_3 < a_1$
      ii. Otherwise, $a_3 < a_2 < a_1$

In tree form:

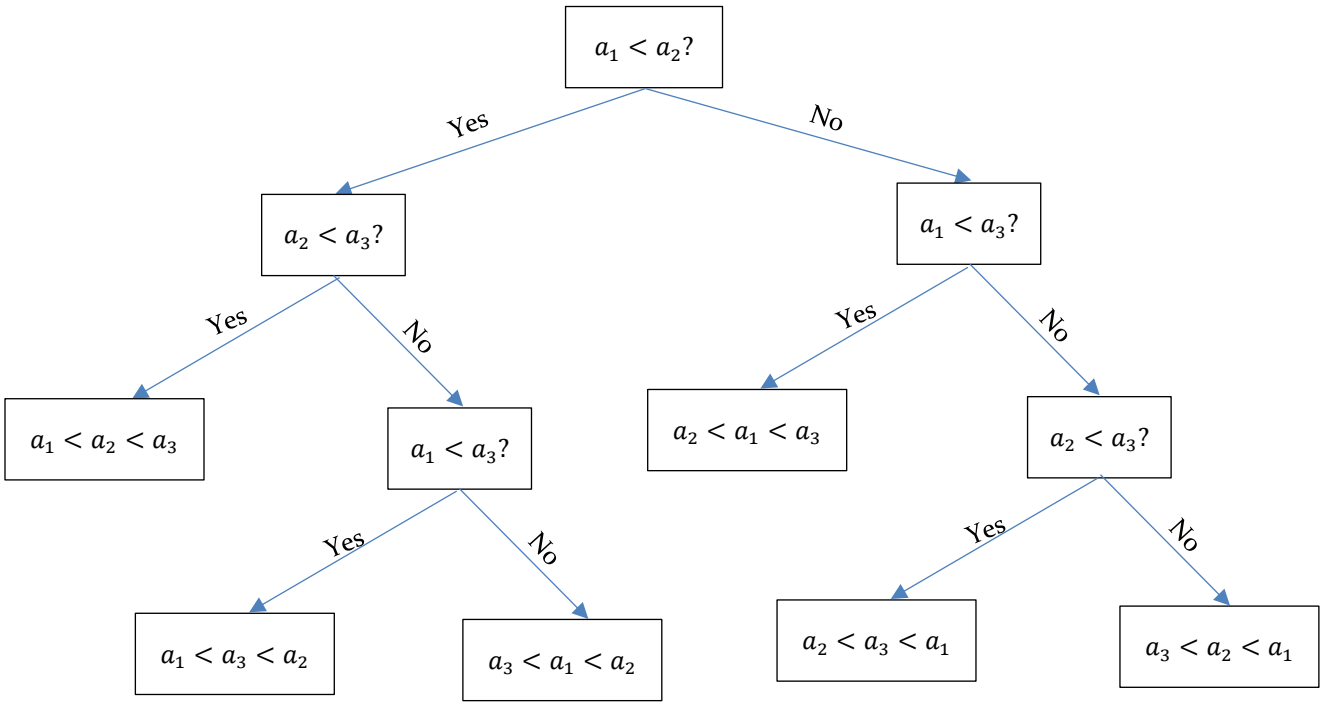


Figure III.B.4.a. A decision tree for sorting three distinct objects

We ended up with a full binary tree with $3! = 6$ leaves (and hence, 5 internal vertices). This is a tree of height $3 = \lceil \log_2 11 \rceil - 1 = \lceil \log_2 6 \rceil$. More generally, to determine the order of n distinct elements, we require a tree with $n!$ leaves and so it must be a tree of height at least $\lceil \log_2(n!) \rceil$.

Observation. For any positive real number r,

$$\log_2 r = \frac{\ln r}{\ln 2}.$$

Observation. For any positive integer n,

$$\log_2(n!) = \sum_{k=1}^{n} \log_2 k = \frac{1}{\ln 2} \sum_{k=1}^{n} \ln k.$$

Observation.

$$\sum_{k=1}^{n} \ln k < \int_{1}^{n+1} \ln x \; dx < \sum_{k=1}^{n} \ln(k+1) = \sum_{k=2}^{n+1} \ln k = \sum_{k=1}^{n+1} \ln k.$$

$$\int_{1}^{n+1} \ln x \, dx = [x \ln x - x]_{1}^{n+1} = (n+1)\ln(n+1) - n.$$

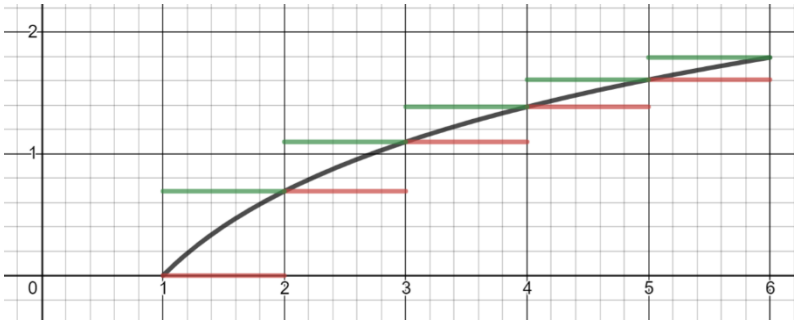


Figure III.B.4.b. Bounding $y = \ln x$ by step functions

Observation.

$$n \ln n - (n-1) < \sum_{k=1}^{n} \ln k < (n+1)\ln(n+1) - n$$

Observation. Binary sorting is $O(n \ln n)$ in the best-case scenario.

III.C. GRAPH SEARCH ALGORITHMS

Suppose the vertices of a graph are labeled with data and we wish to find the location of a specific data item. Two common schemes to search the vertices of a graph are called "depth-first search" and "breadth-first search." We describe these schemes in the subsections to follow.

In each algorithm, we assume G is connected. We also assume there is a mechanism for labelling vertices that have been visited already.

III.C.1. DEPTH-FIRST SEARCH

In a depth-first search, ...

SECTION IV. GRAPH TOPOLOGY

IV.A. PRELIMINARIES

IV.A.1. THE JORDAN CURVE THEOREM

Two questions arose very early in the historical development of graph theory:

1. Could K_5 be drawn without edges crossing?
2. Could $K_{3,3}$ be drawn without edges crossing?

The innocent-sounding phrase "drawn without edges crossing" conceals an enormous amount of deep mathematical thought that took centuries (mostly the 19th and 20th Centuries) to completely hash out. Consider the Jordan curve theorem. This result will be central to our discussion in this section.

Definition. The **unit circle** is the set of points $C = \{(\cos t\,, \sin t)\colon\ 0 \le t < 2\pi\} \subset \mathbb{R}^2$. We choose to parametrize the unit circle this way, so for a map $\phi\colon C \to S$, we can write $\phi(t)$ instead of $\phi(\cos t\,, \sin t)$.

Definition. A **simple closed curve (or Jordan curve)** is the image of a continuous bijection $\phi\colon C \to \mathbb{R}^2$ where C is the unit circle.

Theorem. (The Jordan Curve Theorem) If K is a simple closed curve, then $\mathbb{R}^2 - K$ has exactly two connected components. One of these components (the **interior**) is bounded and the other component (the **exterior**) is unbounded.

This result may seem obvious: If you draw a closed non-self-intersecting curve on a sheet of paper (imagine the sheet is infinite), then the sheet of paper is dissected into two territories, one inside the curve and the other outside the curve. The difficulties in proving this theorem lie mainly in how loosely a Jordan curve is defined. Continuous maps can be extremely complicated.

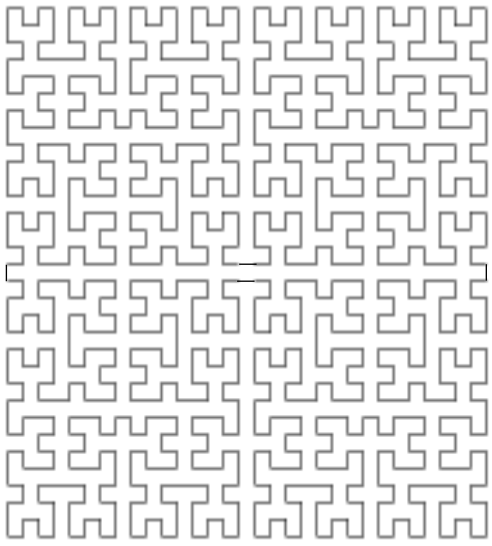


Figure IV.A.2.a. A Jordan Curve

IV.A.2. DRAWINGS OF GRAPHS

The question, "what does it mean to draw a graph?" is related to the Jordan curve theorem in that we use the same language involving continuous bijections. Since we will be considering drawings on surfaces other than $\mathbb{R}^2$, e.g. the unit sphere, we will proceed intuitively as needed rather than developing the considerable theory necessary for a rigorous treatment. For instance, we will assume a surface is a subset of $\mathbb{R}^3$ and treat "continuity" as defined in the usual manner for mappings into $\mathbb{R}^3$.

Definition. A **drawing of a graph G** on a surface S is a set of points Q of S such that

- There exists a bijection $\Upsilon: V \to Q$; this means every vertex corresponds to its own location on Q.
- For each loop $\ell = vv$, there exists a continuous (as a mapping into S) bijection $\Lambda_\ell: C \to Q$ such that $\Lambda_\ell(0) = \Upsilon(v)$. The points of Q corresponding to $\Lambda_\ell(x)$ where $0 < x < 2\pi$ are called **interior** points of ℓ.
- For each edge $e = uv$, there exists a continuous (as a mapping into S) bijection $\mathrm{E}_e: [0,1] \to Q$ such that $E_e(0) = \Upsilon(u)$ and $E_e(1) = \Upsilon(v)$. The points of Q corresponding to $E_e(x)$ where $0 < x < 1$ are called **interior** points of e.
- There is no vertex v, loop ℓ, or edge e such that $\Upsilon(v)$ is an interior point of ℓ or an interior point of e. This means no edge passes through a vertex.

Definition. Given a drawing of G, two edges and/or loops **cross** at a point $p \in Q$ if p an interior point of both edges.

We can treat the drawing Q as a graph using the vertex set $\Upsilon(V)$ and the edge set $\{\Lambda_\ell$:all loops $\ell\} \cup \{\mathrm{E}_e$:all edges $e\}$. We will call this a **drawn graph.** Here, we are being loose with the definition of the edge set as the edges are not unordered pairs of vertices. At this point, we can choose to adjust our definition to accommodate this situation.

Given a set S, we define $(S^2)^*$ as the set of unordered pairs of S.

A **graph** G is a set V of vertices and a set E of edges along with a function $\Psi: E \to (V^2)^*$. In this definition, the edges themselves can be anything. The endpoints of an edge e are the elements of $\Psi(e)$.

All of these definitions provide a formal framework to treat graphs as sets of points on surfaces. For instance, consider the graphs in Figure IV.A.2.a.

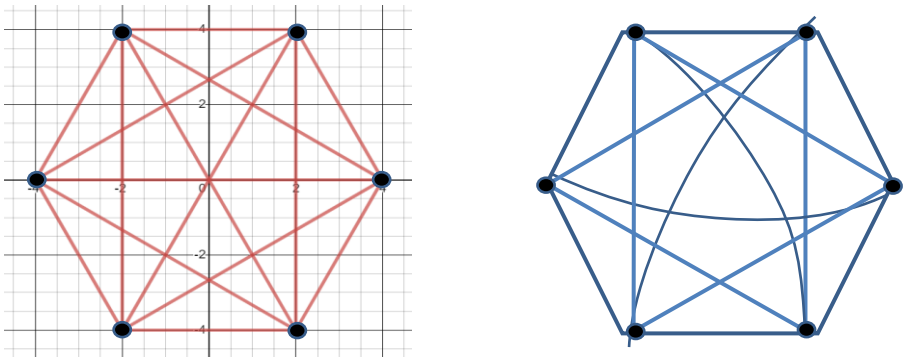


Figure IV.A.2.a. Drawings of K_6 in $\mathbb{R}^2$

The left graph shows a drawing of K_6 where all of the edges are segments. Notice that three of the edges cross at the origin and that there are twelve other points in $\mathbb{R}^2$, e.g. $(2,0)$, where pairs of edges cross. The vertices are at $(4,0), (2,4), (-2,4), (-4,0), (-2,-4)$, and $(2,-4)$.

The graph on the right of Figure IV.A.2.a. depicts a drawing of K_6 where the three radial edges do not all cross at the origin.

Note. There is no requirement that edge be drawn as segments; when all edges are drawn as segments, we call the resulting drawn graph a **segment graph** (not in common use). A graph with loops cannot be a segment graph.

A common technique for dealing with drawings of graphs is to use the following topological properties of subsets of $\mathbb{R}^k$ using the Euclidean distance. Here, $d(a,b)$ is the Euclidean distance between points a and b.

- The Heine-Borel Theorem: In $\mathbb{R}^k$, a set is **compact** if and only if it is closed and bounded.
- Finite sets are compact.
- Given two nonempty compact sets A and B, there exist points $a^* \in A$ and $b^* \in B$ such that $d(a^*, b^*)$ minimizes $\{d(a,b): a \in A, b \in B\}$.
- If A is compact and $f: A \to \mathbb{R}^k$ is continuous, then $f(A)$ is compact.
- The closed interval [0,1] and the unit circle are both compact sets.
- Drawn edges are compact.
- Drawn finite graphs are compact.

IV.B. PLANAR GRAPHS

Planar graphs are those graphs that can be drawn in the plane ($\mathbb{R}^2$) without edges crossing. The problem of determining which graphs are planar and which are not has an extensive literature, pioneered by Euler's work involving the Euler characteristic and culminating in Kuratowski's theorem which gives a concise characterization of planar graphs.

IV.B.1. DEFINITIONS AND PRELIMINARY RESULTS

Definition. A graph G is **planar** if there exists a drawing of G in $\mathbb{R}^2$ where no two edges cross.

Definition (uncommon). A graph G is **segmented** if there exists a drawing of G in $\mathbb{R}^2$ that is a segment graph.

Definition. An actual drawing of a planar graph in $\mathbb{R}^2$ is called a **plane** graph. In this case, the vertices are the points and the edges are the curves in $\mathbb{R}^2$.

Observation. Cycle graphs K_n are segmented planar graphs. This is immediate by considering the vertex set

$$V = \{v_k = (\cos 2\pi k/n\,, \sin 2\pi k/n)\colon\ k \in \mathbb{Z}, 0 \le k \le n-1\}$$

and where two vertices v_i, v_j are adjacent if and only if $j - i = 1 \bmod n$ or $i - j = 1 \bmod n$. If we join adjacent vertices with segments, the result is a regular n-gon.

Theorem IV.B.1.a. Rooted trees are segmented planar graphs.

Sketch of Proof. Let T, r be a rooted tree. Place the root r at the origin. Index the neighbors of r, $w_0, w_1, \ldots, w_{q-1}$ and draw them at q equally distributed points on the unit circle together with the edges joining them to r as radii of the circle. For each vertex w_i at level 1, draw a circle centered at w_i of sufficiently small positive radius that contains no part of the graph drawn so far, except for part of the edge $e_i = rw_i$. A compactness argument suffices to show that this circle exists. Draw the neighbors of w_i equally distributed around that circle, avoiding the edge e_i. Continue this process for each level until the tree is drawn.

□

Corollary IV.B.1.b. Forests are segmented planar graphs.

Definition. A graph is **spherical** if it can be drawn on the surface of a sphere without edges crossing.

We can map the plane to the surface of the sphere in a nearly one-to-one fashion—the north pole of the sphere is the only point not matched—as illustrated in Figure IV.B.1.c. Here, we illustrate how to project each point of the plane $z = 0$ to a point on the unit sphere $x^2 + y^2 + z^2 = 1$.

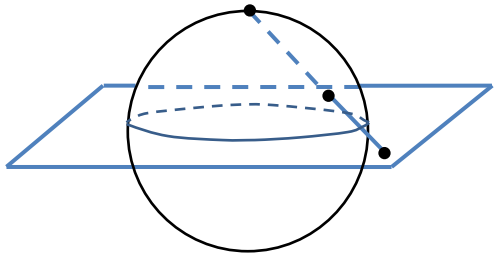


Figure IV.B.1.c. Injectively mapping the plane to the unit sphere.

As a consequence, planar graphs are also spherical. Also, any spherical graph with no vertex or edge at the north pole is planar.

We offer an intuitive definition that will play a significant role as we move forward.

Definition. Two sets A and B of $\mathbb{R}^n$ are **homeomorphic** if there exists a continuous bijection $f: A \to B$.

Figure IV.B.1.c suggests that the plane and the punctured unit sphere are homeomorphic. A formal proof of this fact is beyond the scope of this text.

Definition. A subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ is called a **region** if it is homeomorphic to the open unit disk

$$D = \{(x, y): x^2 + y^2 < 1\}.$$

Observation. If S is the unit sphere then $R = S - \{(0,0,1)\}$ is a region. We see this by using spherical coordinates:

$$R = \left\{(\cos\theta\cos\phi\,, \sin\theta\cos\phi\,, \sin\phi):\ 0 \le \theta < 2\pi, -\frac{\pi}{2} \le \phi < \frac{\pi}{2}\right\}$$

and the mapping $f: R \to D$ given by

$$f(\cos\theta\cos\phi\,, \sin\theta\cos\phi\,, \sin\phi) = \left(\frac{1}{\pi}\left(\phi + \frac{\pi}{2}\right)\cos\theta\,, \frac{1}{\pi}\left(\phi + \frac{\pi}{2}\right)\sin\theta\right).$$

Graphically, we map curves of constant longitude on the sphere to radii of the open unit disk. The south pole is mapped to the center of the disk. See Figure IV.B.1.d.

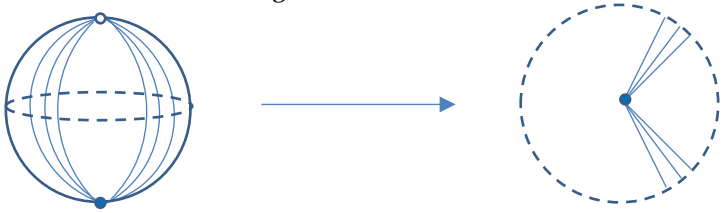


Figure IV.B.1.d. Mapping the punctured sphere to the open disk.

Observation. If G is planar, then any subgraph H of G is planar. This follows immediately by drawing G in the plane and then deleting edges and or vertices to obtain H.

Observation. If G is not planar, then any graph H that contains G as a subgraph is not planar.

IV.B.2. EULER'S FORMULA FOR SPHERICAL (AND HENCE, PLANAR) CONNECTED GRAPHS

We consider only connected graphs in our discussion of Euler's formula. We state the following without proof:

Proposition IV.B.2.a. If a connected finite spherical graph G is drawn on the sphere S without edges crossing, then the set $S - G$ is the disjoint union of a finite number of regions; this number is denoted r.

Proposition IV.B.2.b. An edge is on the boundary of either one or two regions. Furthermore, an edge is on the boundary of one region if and only if it is a bridge.

Proposition IV.B.2.c. A region bounded by non-bridges, exclusively, is bounded by the edges in a cycle.

Observations.

- If G is simple, its non-bridge bound regions are bounded by cycles with at least three edges.
- If G is simple and bipartite, its non-bridge bound regions are bounded by cycles with at least four edges.

Corollary IV.B.2.d. The graph T is a tree drawn on a sphere without edges crossing iff $S - T$ consists of a single region.

Euler's formula relates the number of vertices, edges, and regions of a graph drawn on a sphere.

Theorem IV.B.2.e (Euler's Formula). For any connected spherical finite graph G drawn on a sphere,

$$n - m + r = 2.$$

Proof. We proceed by induction on $r \geq 1$, the number of regions. For the base case, suppose G is a graph where $r = 1$. By corollary IV.B.2.d, G is a tree and so $m = n - 1$. Hence,

$$n - m + r = n - (n - 1) + 1 = 2$$

as desired.

For the inductive step, we assume the result holds for $r = k \geq 1$, i.e., Euler's formula holds for any graph with k regions. Let G be a graph with $r = k + 1$ regions. Since $r \geq 2$, G is not a tree and so it must have non-bridges. Let e be a non-bridge. Then e is on the boundary of two regions R_1 and R_2. The graph $G' = G - e$ can be drawn by absorbing the points of e back onto the sphere. This has the effect of merging R_1 and R_2 into a single region for G'. More precisely, $R_1 \cup e \cup R_2$ is a single region for G'. We then observe that

$$n' = n, m' = m - 1, r' = r - 1 = k$$

and so

$$n - m + r = n' - (m' + 1) + (r' + 1) = n' - m' + r' = 2$$

where the last equality is by induction because G' has $r' = k$ regions.

□

We note that when a spherical graph is projected onto the plane (in the sense of figure IV.B.1.c), the region that contains the north pole is projected onto an unbounded set of the plane. All other regions are projected onto bounded regions of the plane. One formulation of Euler's formula for planar graphs:

Theorem IV.B.2.f (Euler's Formula for the Plane). For any connected finite graph G drawn on the plane,

$$n - m + r = 2,$$

where r is the number of regions, including a unique unbounded region, into which G separates the plane.

Corollary IV.B.2.g (Euler's Formula for Convex Polyhedra). If P is a convex polyhedron with n vertices, m edges, and r faces, then $n - m + r = 2$.

Sketch of Proof. Choose an interior point x of P and a bounding sphere for P. Project P from x onto the sphere; this results in P being drawn as a spherical graph whose vertices, edges, and regions are in bijection with the vertices, edges, and faces of P.

□

IV.B.3. THE TOTAL EDGE COUNT

We now consider counting the edges around a region; this will be analogous to the degree of a vertex.

Definition. Given a region R, the **edge count** of R is the number of edges on the boundary of R, counting bridges twice.

This is consistent with traversing the edges around the boundary of R, noting that a bridge is encountered twice during such a traversal.

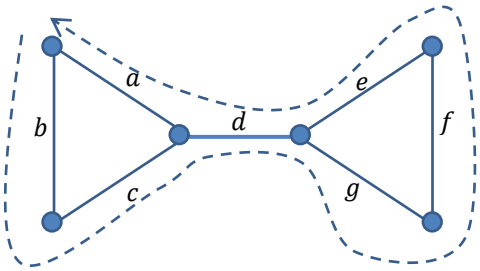


Figure IV.B.3.a. Traversing the edges of the "external" region R.

In Figure IV.B.3.a, traversing the edges counterclockwise starting at the upper left vertex produces the edge sequence

$$b, c, d, g, f, e, d, a$$

and so the edge count of R is 8. We note that edge d is a bridge and is counted twice.

Definition. The **total edge count** of a graph, denoted ε, drawn on the sphere (or plane) is the sum of the edge counts of the regions for that graph.

Observations.

- Since each bridge contributes 2 to the edge count of its region and each non-bridge contributes 1 to the edge count of each of two regions, every edge contributes 2 to the total edge count.
- The total edge count is twice the number of edges, i.e., $\varepsilon = 2m$, and so ε is necessarily even.

The total edge count will play a central role in many results. We start with elementary proofs that neither of the graphs K_5 or $K_{3,3}$ is planar.

Theorem IV.B.3.b. The graph K_5 is not planar.

Proof. Suppose otherwise and suppose we have a plane drawing of K_5. Such a drawing must have $n = 5$ vertices, $m = 10$ edges, and from Euler's formula, $r = 2 - n + m = 2 - 5 + 10 = 7$ regions. Each region must have an edge count of at least 3, and so the total edge count, ε, must satisfy $\varepsilon \geq 3 \cdot 7 = 21$. More precisely, since ε is even, we have $\varepsilon \geq 22$. On the other hand, $\varepsilon = 2m = 20$, leading to the conclusion that $20 = \varepsilon \geq 22$ which cannot happen.

□

Theorem IV.B.3.c. The graph $K_{3,3}$ is not planar.

Proof. Suppose otherwise and suppose we have a plane drawing of K_5. Such a drawing must have $n = 6$ vertices, $m = 9$ edges, and from Euler's formula, $r = 2 - n + m = 2 - 6 + 9 = 5$ regions. Since $K_{3,3}$ is bipartite, each region must have an edge count of at least 4, and so the total edge count, ε, must satisfy $\varepsilon \geq 4 \cdot 5 = 20$. On the other hand, $\varepsilon = 2m = 18$, leading to the conclusion that $18 = \varepsilon \geq 20$ which cannot happen.

▫

Definition. A planar graph is **triangulated** if all of its regions are triangles, i.e., each is bounded by three distinct edges.

Definition. If R is a region bounded by $k \geq 4$ distinct edges, then a new edge through R that joins nonconsecutive vertices of the boundary of R is called a **chord.**

Proposition IV.B.3.d. Any region bounded by $k \geq 3$ distinct edges can be dissected into triangles by adding chords with none of the chords crossing.

Corollary IV.B.3.e. If G is planar with no bridges or parallel edges, then G can be triangulated by adding chords.

Theorem IV.B.3.f. For any triangulated planar graph, $m = 3n - 6$.

Proof. Let G be triangulated with r regions. Then the total edge count is $\varepsilon = 3r = 2m$. By Euler's formula,

$$\begin{aligned} n - m + r &= 2 \\ 3n - 3m + 3r &= 6 \\ 3n - 3m + 2m &= 6 \\ 3n - m &= 6 \\ m &= 3n - 6 \end{aligned}$$

as desired.

▫

Corollary IV.B.3.g. For any planar graph without bridges or parallel edges, $m \leq 3n - 6$.

Corollary IV.B.3.h. The graph K_5 is not planar.

Proof. We have $m = 10 > 9 = 3(5) - 6 = 3n - 6$ in contrast to the previous corollary.

Now, consider the p-regular planar graphs G where all regions have the same edge count $k \geq 3$. Let n be the number of vertices as usual. From total degree and total edge count considerations,

$$pn = 2m = kr$$

From Euler's formula,

$$\begin{aligned} n - m + r &= 2 \\ pkn - pkm + pkr &= 2pk \end{aligned}$$

$$2km - pkm + 2pm = 2pk$$
$$2m(p + k) = (2 + m)pk$$
$$p + k = \left(\frac{1}{m} + \frac{1}{2}\right)pk$$

Now, suppose $p = 2$. Then

$$2 + k = \left(\frac{1}{m} + \frac{1}{2}\right)2k = \frac{2k}{m} + k \text{ and so } 2m = 2k.$$

This is the case when G is a cycle.

If $p = 3$, then

$$3 + k = \left(\frac{1}{m} + \frac{1}{2}\right)3k$$
$$3 = \frac{3k}{m} + \frac{k}{2} > \frac{k}{2}.$$

This forces $k < 6$. We then have $3 \leq k \leq 5$. If $k = 3$, then

$$3 = \frac{9}{m} + \frac{3}{2}; \frac{3}{2} = \frac{9}{m}; m = 6.$$

This leads to $n = \frac{2m}{p} = \frac{12}{3} = 4$ and $r = \frac{2m}{k} = \frac{12}{3} = 4$.

If $k = 4$, then

$$3 = \frac{12}{m} + 2; m = 12.$$

Here, $n = \frac{2m}{p} = \frac{24}{3} = 8$ and $r = \frac{2m}{k} = \frac{24}{4} = 6$.

If $k = 5$, then

$$3 = \frac{15}{m} + \frac{5}{2}; m = 30.$$

Here, $n = \frac{2m}{p} = \frac{60}{3} = 20$ and $r = \frac{2m}{k} = \frac{60}{5} = 12$.

In Figure IV.B.3.i, we show planar graphs that correspond to these values:

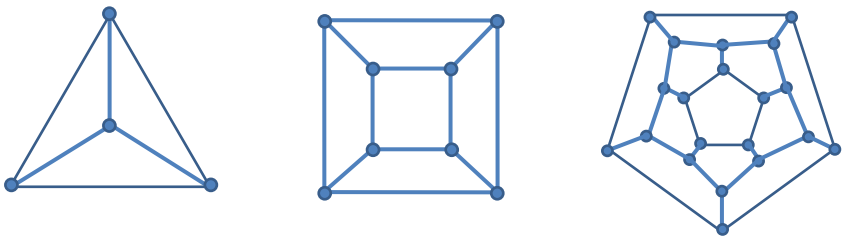


Figure IV.B.3.i. Examples of 3-regular planar graphs with $k = 3,4,5$.

If $p = 4$, then

$$4 + k = \left(\frac{1}{m} + \frac{1}{2}\right)4k$$
$$4 + k = \frac{4k}{m} + 2k$$
$$4 = \frac{4k}{m} + k > k$$

and so $k = 3$ is the only possibility. This leads to

$$4 + 3 = \frac{12}{m} + 6, \text{ or } m = 12;\ n = \frac{2m}{p} = \frac{24}{4} = 6 \text{ and } r = \frac{2m}{k} = \frac{24}{3} = 8.$$

If $p = 5$, then

$$5 + k = \left(\frac{1}{m} + \frac{1}{2}\right)5k$$
$$5 + k = \frac{5k}{m} + \frac{5k}{2}$$
$$5 = \frac{5k}{m} + \frac{3k}{2} > \frac{3k}{2}$$
$$10 > 3k$$

and, again, $k = 3$ is the only possibility, leading to

$$5 + 3 = \frac{15}{m} + \frac{15}{2}, or\ m = 30; n = \frac{2m}{p} = \frac{60}{5} = 12 \text{ and } r = \frac{2m}{k} = \frac{60}{3} = 20.$$

In Figure IV.B.3.j, we show planar graphs that correspond to $p = 4,5$ and $k = 3$:

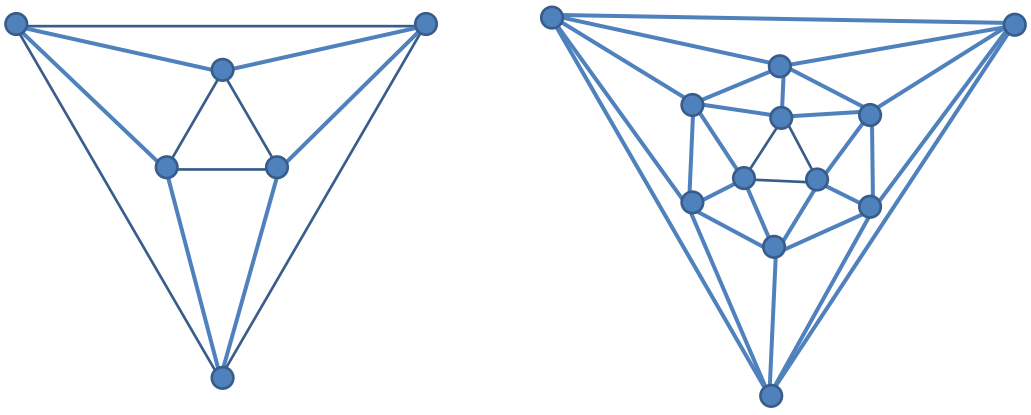


Figure IV.B.3.i. Examples of p-regular planar graphs with $p = 4,5$ and $k = 3$.

These computations are at the heart of showing that there are exactly five **regular polyhedra**, i.e., convex polyhedra where each face of one is the same regular polygon and where the same number of these regular polygons meet at every vertex.

IV.C. GRAPHS ON OTHER SURFACES

IV.C.1. REPRESENTATIONS OF SURFACES

A formal discussion of possible surfaces embedded in $\mathbb{R}^3$ is beyond the scope of this text. Instead, we proceed intuitively, using the sphere as our starting point. One useful construction for surfaces involves "surgery" where we cut along a curve segment and identify points with the result. For instance, with a sphere, we can cut along a longitude from the north pole to the south pole, and then "open it up and flattening it" as depicted in Figure IV.C.1.a.

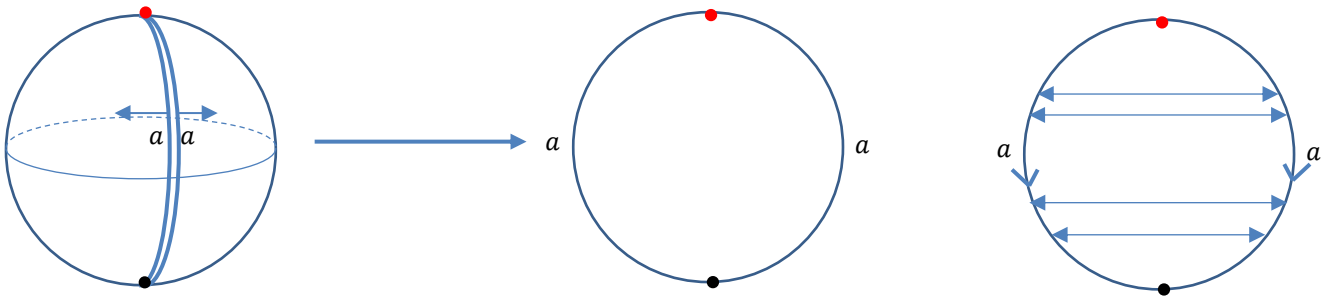


Figure IV.C.1.a. Cutting a Sphere and Flattening Out the Result.

Notice that the curve a is oriented with the red point (the north pole) at the top and the black point (the south pole) at the bottom. A common representation is to place small arrows on the curve representing a, as in the far right of Figure IV.C.1.a. This specifically tells us that the points along the left version of the curve a are identified precisely with the corresponding points along the right version of the curve a.

In contrast, Figure IV.C.1.b shows the disk with the curves a oriented in opposite directions. This causes the points along a to be identified differently; one side effect of this is that the north pole and the south pole of this disk are actually the same point. This surface, known as the **projective plane**, is wildly different from a sphere; we will see some of these differences in later sections.

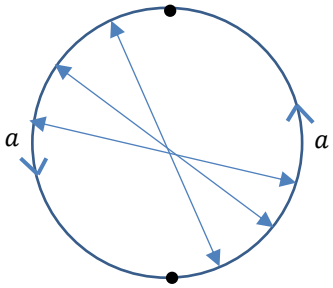


Figure IV.C.1.b. A Representation of the Projective Plane

We now consider the surface of a donut, called a **torus**. In Figure IV.C.1.c, we see how to perform "surgery" on a torus to produce the representation on the right where parallel sides are identified in the orientations shown.

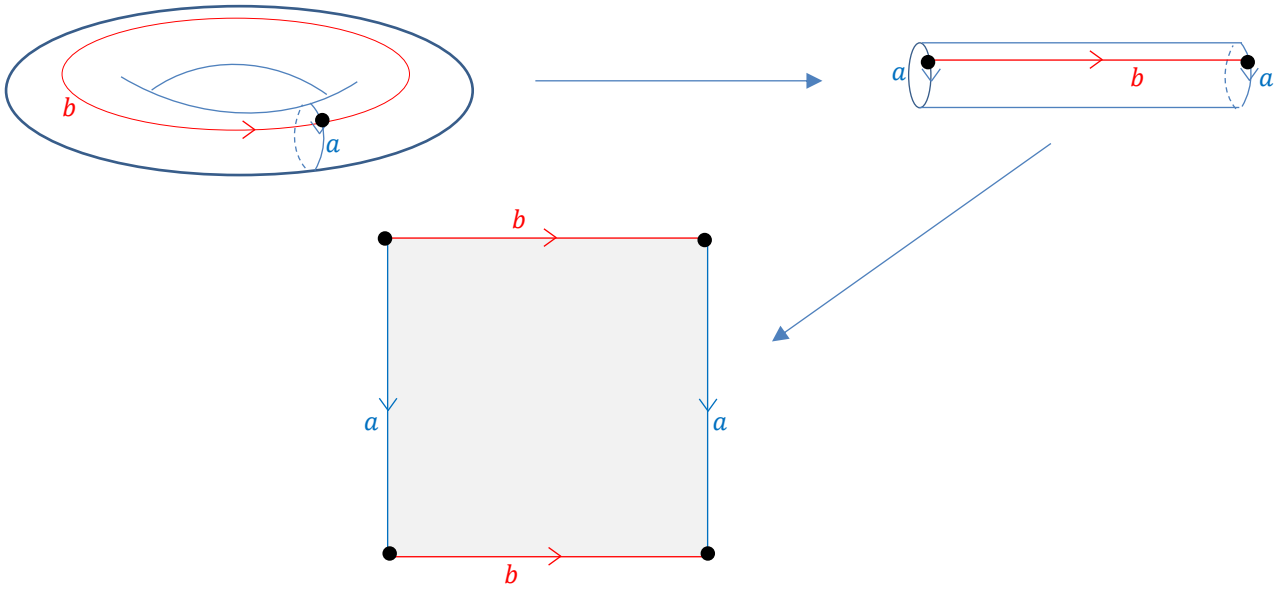


Figure IV.C.1.c. Surgery on a Torus

In the final figure, we note that all four of the corners are identified; they represent the same point on the surface. If we traverse the arcs clockwise starting from the upper left corner, we obtain the representation $aba^{-1}b^{-1}$, where a^{-1} and b^{-1} indicate traversing the arcs a and b against their orientations. We declare any traversal in any direction starting at any corner to represent the same surface. These include

$$aba^{-1}b^{-1}, ab^{-1}a^{-1}b, bab^{-1}a^{-1}, ba^{-1}b^{-1}a, a^{-1}bab^{-1}, a^{-1}b^{-1}ab, b^{-1}aba^{-1}, b^{-1}a^{-1}ba.$$

We note that if we do not assume multiplication is commutative, and we start with the equation $ab = ba$, then we can arrive at any of the strings in the list above equaling e, the identity element. For instance,

$$\begin{aligned} ab &= ba \\ b^{-1}ab &= b^{-1}ba \\ b^{-1}ab &= a \\ b^{-1}aba^{-1} &= aa^{-1} \\ b^{-1}aba^{-1} &= e \end{aligned}$$

Algebraically (using the traversal interpretation) the sphere can be represented as aa^{-1} and the projective plane can be represented as aa.

By rearranging pieces and reuniting identified edges, surfaces can be shown to be equivalent. In Figure IV.C.1.d, we show how to rearrange $aba^{-1}b^{-1}$ into $cdec^{-1}d^{-1}e^{-1}$.

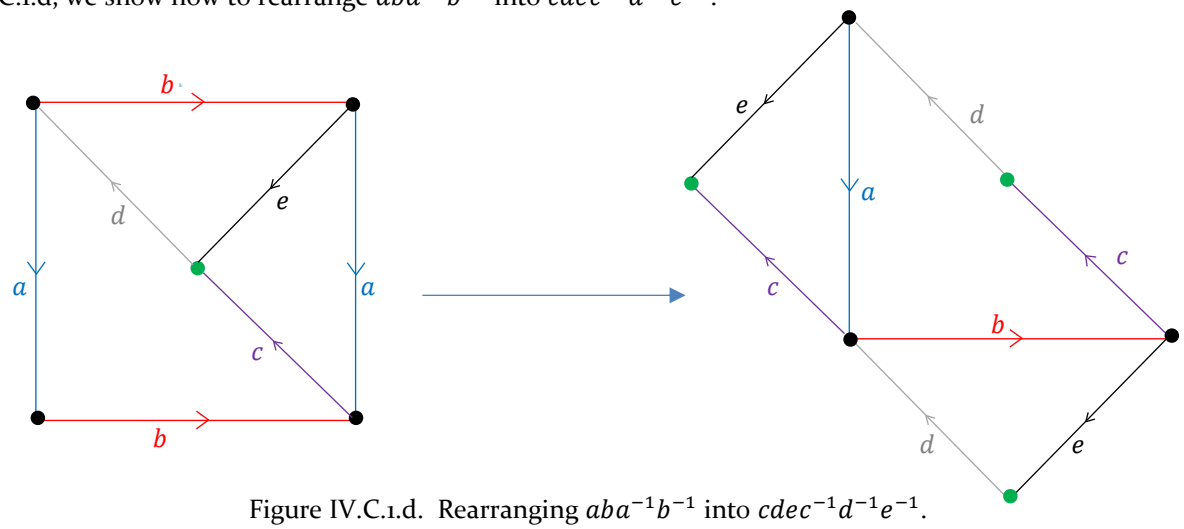


Figure IV.C.1.d. Rearranging $aba^{-1}b^{-1}$ into $cdec^{-1}d^{-1}e^{-1}$.

Moving on to more exotic surfaces, we consider the "two-holed torus" depicted in Figure IV.C.1.e, wherein we suggest a way to knit together two one-holed tori (the plural of torus is "tori") to form a two-holed torus.

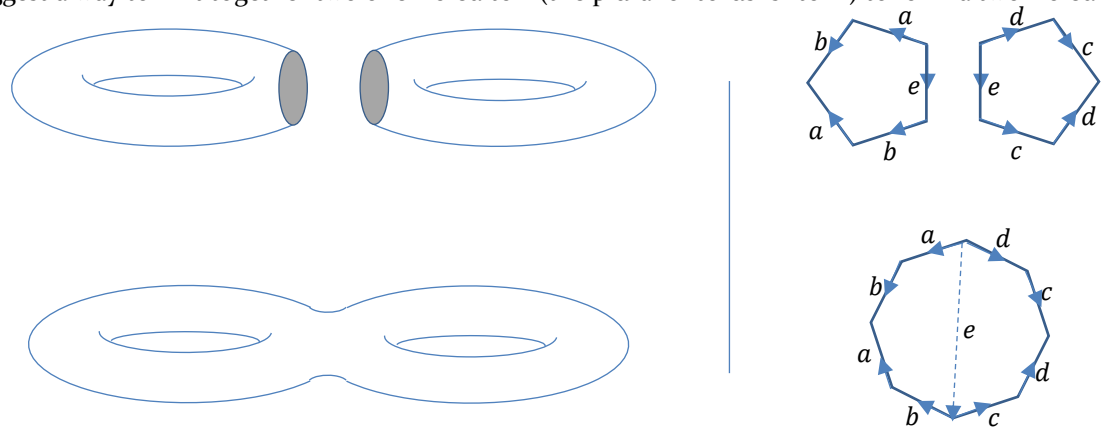


Figure IV.C.1.e. Knitting together two tori to form a two-holed torus.

In Figure IV.C.1.e, the e arcs represent the boundaries of the holes that would be identified to form the two-holed torus. Algebraically, the two-holed torus has a boundary represented by the expression

$$aba^{-1}b^{-1}cdc^{-1}d^{-1}.$$

More generally, we offer (without proof) the following results:

Proposition. The algebraic expressions

$$a_1b_1a_1^{-1}b_1^{-1}\,a_2b_2a_2^{-1}b_2^{-1}\cdots a_nb_na_n^{-1}b_n^{-1}, \qquad a_1a_2\cdots a_{2n}a_1^{-1}a_2^{-1}\cdots a_{2n}^{-1},$$
$$a_1a_2\cdots a_{2n}a_{2n+1}a_1^{-1}a_2^{-1}\cdots a_{2n}^{-1}a_{2n+1}^{-1}$$

all represent the boundary of an n-holed torus T_n. The corners of the $4n$-gon bounded by each of the first two expressions all represent the same point, while those of the $(4n+2)$-gon represent two distinct points.

Proposition. Every orientable surface is homeomorphic (continuously deformable) to an n-holed torus.

For non-orientable surfaces, we have the following result:

Proposition. The algebraic expression $a_1a_1a_2a_2 \ldots a_ka_k$ represents the boundary of a non-orientable surface P_k. Recall that the projective plane is P_1. The corners of the $2k$-gon bounded by this expression all represent the same point.

Bringing these results together, we have

Proposition (Classification of Surfaces Theorem). Every compact (recall this means "closed and bounded" in $\mathbb{R}^n$) surface embedded in $\mathbb{R}^3$ has a boundary that is algebraically represented by the concatenation of T_n with P_k for some $n \geq 0, k \geq 0$. If both $n = k = 0$, then the surface is a sphere.

For details of a proof, see: Francis, George K.; Weeks, Jeffrey R. (May 1999), "Conway's ZIP Proof" (PDF), *American Mathematical Monthly*, **106** (5): 393

IV.C.2 EULER'S FORMULA FOR GRAPHS ON OTHER SURFACES

We now generalize Euler's formula for graphs drawn on other surfaces. We concentrate on graphs drawn on T_n and P_k separately.

In what follows, we assume that "G is drawn on a surface S" means that $S - G$ is a finite collection of regions; this implies G is connected. Some graphs cannot be drawn on some surfaces; for instance, a tree can be drawn only on the sphere.

Observation. If G is drawn on a surface S so that $S - G$ is a finite collection of regions and if e is on the boundary of two regions, then deleting e from the drawing of G decreases the number of edges by 1 and the number of regions by 1 (because the two regions touching e are merged into one.) Hence, if $G' = G - e$ is the resulting drawn graph on S, we have

$$n - m + r = n' - m' + r'.$$

Observation. If G is drawn on a surface S and v is a leaf, then we can delete v without changing the number of regions. This has no effect on the quantity $n - m + r$.

Observation. If G is drawn on a surface S with a vertex v of degree 2, then the two edges incident on v can be merged.

In short, the quantity $n - m + r$ is invariant under the operations described in these observations.

We state without proof the following result:

Proposition IV.C.2.a. These operations can be performed until we reach one vertex and one region; the resulting graph always has the same number of edges.

Definition. Given a surface S, there is a quantity $\chi(S)$ called the **Euler characteristic of S,** such that every graph drawn on S that dissects S into regions satisfies

$$n - m + r = \chi(S).$$

Example IV.C.2.b. The Euler characteristic of the sphere T_0 is $\chi(T_0) = 2$, from Euler's formula $n - m + r = 2$.

Example IV.C.2.c. The Euler characteristic of the one-holed torus T_1 is $\chi(T_1) = 0$. We can verify this fairly quickly using the diagram in Figure IV.C.2.d.

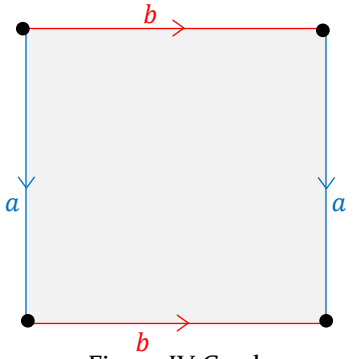


Figure IV.C.2.d.

If we treat the corner point (recall all four corners of this figure represent the same point) as a single vertex and each of the pairs of sides labeled a and b as an edge, the central territory is a region and so we have $n = 1, m = 2, r = 1$ and

$$n - m + r = 1 - 2 + 1 = 0.$$

Since this is invariant for all graphs drawn on T_1, $\chi(T_1) = 0$.

Example IV.C.2.d. For an h-holed torus, the same construction as for T_1 yields $n = 1$ (the corners are all the same point), $m = 2h$ (each hole accounts for two edges a_h and b_h), and $r = 1$. Thus,

$$n - m + r = 1 - 2h + 1 = 2 - 2h$$

and so $\chi(T_h) = 2 - 2h$.

Example IV.C.2.e. For an h-fold projective plane S_h, we have $n = 1, m = h, r = 1$ and

$$n - m + r = 1 - h + 1 = 2 - h,$$

implying $\chi(S_h) = 2 - h$.

Observation. If the Euler characteristic is odd, then the surface is not orientable.

Definition. Given an orientable surface S, its **genus** $g(S)$ satisfies $\chi(S) = 2 - 2g(S)$. Similarly, given a connected graph G, its **genus** $\boldsymbol{g(G)}$ is the smallest genus of an orientable surface on which G can be drawn without edges crossing.

Observations.

- The genus of T_h satisfies $g(T_h) = h$. The genus counts the number of holes in an h-holed torus.
- Since neither K_5 nor $K_{3,3}$ are planar, we have $g(K_5) \geq 1$ and $g(K_{3,3}) \geq 1$.
- If G can be drawn on a surface of genus g, then it can be drawn on a surface of genus $g + 1$, provided we remove the condition that the surface must then be dissected into regions.

For the last observation, raising the genus by 1 is equivalent to putting a "handle" on the surface, as depicted in Figure IV.C.2.f. If G can be drawn on the surface, the handle can be attached without disturbing G.

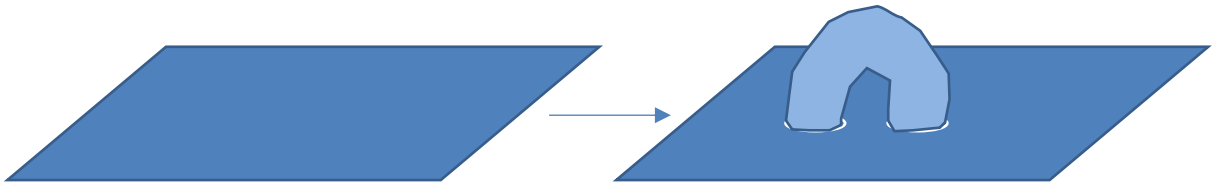


Figure IV.C.2.f. Attaching a handle increases the genus by one.

This process of attaching handles can also be used to eliminate edge crossings of a graph. We draw the handle to accommodate one edge "passing under" the handle while the other edge is drawn on the handle. See Figure IV.C.2.g.

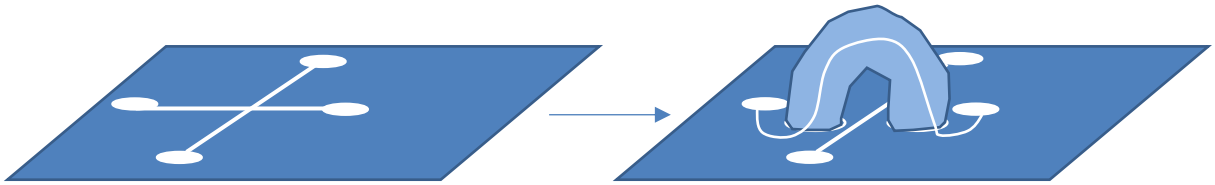


Figure IV.C.2.g. Attaching a handle reduces edge crossings by at least one.

Thus, if G is drawn on a surface of genus g with k edge crossings, then G can be drawn on a surface of genus $g + k$ with no edge crossings.

Example IV.C.2.h. We claim $g(K_5) = 1$. It suffices to draw K_5 with one edge crossing. See Figure IV.C.2.i.

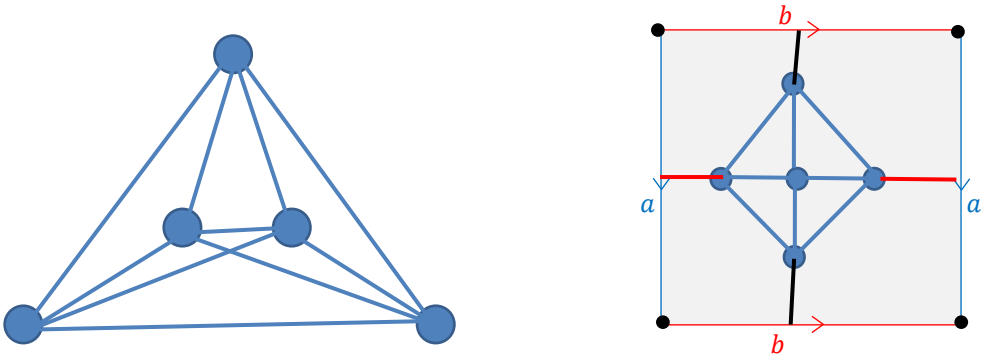


Figure IV.C.2.i. The graph K_5 drawn with a single edge crossing and drawn on T_1.

We also depict in Figure IV.C.2.i the graph K_5 drawn with no edges crossing; here, the red segments together form a single edge as do the black segments taken together.

Example IV.C.2.j. Just because a graph can be drawn on T_1 does not mean it can be drawn with only one edge crossing. We consider K_6 which actually requires three edge crossings even though it can be drawn on T_1 with no crossings; the minimum of three crossings will not be proved here. See Figure IV.C.2.k.

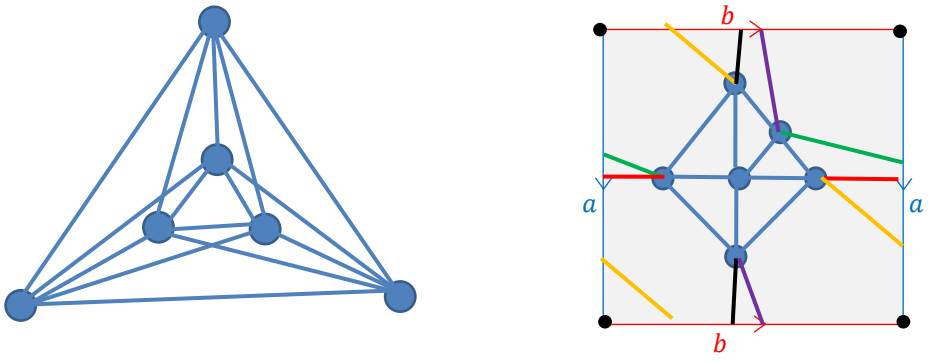


Figure IV.C.2.k. The graph K_6 drawn with three edge crossings and drawn on T_1.

SECTION V. DIRECTED GRAPHS (DIGRAPHS)

V.A. PRELIMINARIES

V.A.1. DEFINITIONS

We start with the terminology of directed graphs, or "digraphs". We alter the wording slightly to distinguish when we are discussing (undirected) graphs from when we are discussing digraphs.

Definition. A **directed graph** (or **digraph**) is a pair (N, A) consisting of a set N of **nodes** and a multiset A of **arcs**, each arc being an ordered pair of nodes.

Notation. We will write $a = uw$ as a shorthand for $a = (u, w) \in A$ with $u, w \in N$.

Conventionally, an arc $a = uw$ is drawn as an arrow as depicted in Figure V.A.1.a.

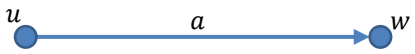


Figure V.A.1.a. The arc $a = uw$.

Definition. Two arcs $a = uw$ and $b = uw$ are said to be **parallel** while two arcs $a = uw$ and $b = wu$ are said to be **anti-parallel** or **opposite.** An arc wu is an **opposite arc** of an arc uw.

Definition. An arc $a = uu$ is called a **loop**. Loops are their own opposite arcs.

There are many applications where one wants to consider only the graph structure of some given digraph, or where one wants to impose a digraph structure on some given graph. In Figure V.A.1.b, we show a graph G and a digraph D where G is the underlying graph of D and D is an orientation of G.

Figure V.A.1.b. A graph G and a digraph D where G is the underlying graph of D

Definition. Given a digraph $D = (N, A)$, its **underlying graph** is the graph $G = (V, E)$ such that $V = N$ and where $\mu_E(uv) = \mu_A(uv) + \mu_A(vu)$. Intuitively, the underlying graph is obtained by "erasing the arrowheads" from the arcs of D, turning them into edges of G.

Definition. Given a graph $G = (V, E)$, an **orientation** of G is a digraph $D = (N, A)$ such that $N = V$ and where each edge $e = uv$ corresponds to exactly one arc a where $a = uv$ or $a = vu$; again, we want $\mu_E(uv) = \mu_A(uv) + \mu_A(vu)$ to hold. Intuitively, an orientation is obtained by "placing arrowheads" on the edges of G, turning them into arcs of D.

Another way to transform a graph $G = (V, E)$ into a digraph $D = (N, A)$ is by declaring $N = V$, and then treating every loop $uu \in E$ as a loop $uu \in A$ and replacing every non-loop $uv \in E$ with a pair of opposite arcs $uv \in A$ and $vu \in A$.

In Figure V.A.1.c, we perform this operation on the graph G depicted:

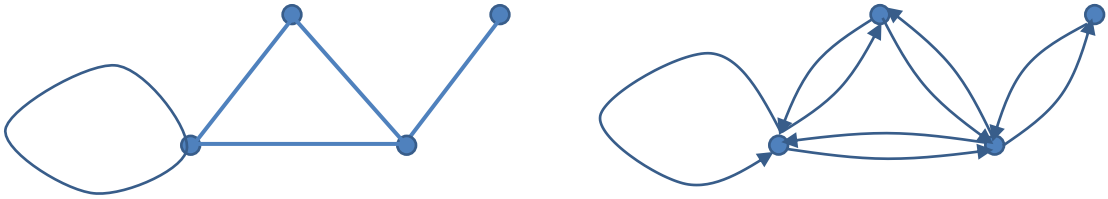


Figure V.A.1.c. Transforming the graph G into a digraph D

The notions of subgraph and induced subgraph carry over to directed graphs without alteration, except to use the terms “node” and “arc” instead of “vertex” and “edge.”

V.A.2. ADJACENCY, INCIDENCE, IN-DEGREE, AND OUT-DEGREE

We can adapt much of the terminology from undirected graphs to fit digraphs. Adjacency and incidence carry over directly:

Definitions. Each of the following is equivalent to $a = uv \in A$ or $a = vu \in A$:

- u and v are **adjacent nodes**
- u is **adjacent** to v and v is **adjacent** to u
- u is **incident** with a and v is **incident** with a
- a is **incident** with u and a is **incident** with v
- u and v are the **end nodes** or **endpoints** of a
- u and v are adjacent in the underlying graph
- Each of u and v is incident with the edge corresponding to a in the underlying graph

To capture the direction of the arc in terms of its end nodes, we have the following refinements:

Definitions. Each of the following is equivalent to $a = uv \in A$:

- u is the **tail** and v is the **head** of a
- u **precedes** v
- v **follows** u
- a **goes from** u **to** v

We refine the concept of degree in digraphs to capture how often a node is a tail and how often it is a head:

Definitions.

- The **in-degree** of a node $u \in N$ is the cardinality of the multiset $\{wu : wu = a \in A\}$, i.e., the number of arcs a having u as the head of a. This is denoted $d_{\text{in}}(u)$.
- The **out-degree** of a node $u \in N$ is the cardinality of the multiset $\{uw : uw = a \in A\}$, i.e., the number of arcs a having u as the tail of a. This is denoted $d_{\text{out}}(u)$.
- The **total in-degree** is the quantity $\sum_{u \in N} d_{\text{in}}(u)$ and the **total out-degree** is the quantity $\sum_{u \in N} d_{\text{out}}(u)$.
- The **minimum in-degree** is $\delta_{\text{in}} = \min_{u \in N} d_{\text{in}}(u)$; the **minimum out-degree** is $\delta_{\text{out}} = \max_{u \in N} d_{\text{out}}(u)$.
- The **maximum in-degree** is $\Delta_{\text{in}} = \max_{u \in N} d_{\text{in}}(u)$; the **maximum out-degree** is $\Delta_{\text{out}} = \max_{u \in N} d_{\text{out}}(u)$.

Observation. A loop contributes 1 to the in-degree and out-degree of the vertex it is incident with.

Observation (The Handshaking Lemma for Digraphs). Since each arc contributes 1 to each of $\sum_{u \in N} d_{\text{in}}(u)$ and $\sum_{u \in N} d_{\text{out}}(u)$, the total in-degree is equal to the total out-degree. Furthermore, both quantities are equal to the number of arcs.

V.A.3. DIRECTED WALKS, THEIR REFINEMENTS, AND STRONG CONNECTEDNESS

We adapt the definitions of walk, path, and cycle to directed graphs:

Definition. Given a digraph $D = (N, A)$, a **directed walk** W is a finite sequence alternating between nodes and arcs that begins and ends at nodes, i.e.,

$$W: v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$$

and such that for all $k, 1 \leq k \leq n$, $e_k = v_{k-1}v_k$; also, if $v_{k-1} = v_k$, then e_k is a loop joining v_k to itself.

Again, $n = |W|$ is the **length** of the walk; this is the number of arcs along the walk.

Definition. A **directed path** is a directed walk where all nodes are distinct.

Definition. A **directed cycle** is a directed walk with $n \geq 1$ such that the only repeated node is $v_n = v_0$.

While there is little connection between a longest directed path in a digraph and a longest path in its underlying graph, there is a surprising connection between lengths of directed paths in a digraph and the chromatic number of its underlying graph.

Theorem V.A.3.a. (Roy (1967) and Gallai (1968), independently) Given a digraph D and its underlying graph G, there exists a directed path of length $\chi(G) - 1$.

Proof. Remove the smallest possible set of arcs A' from D so that $D - A'$ has no directed cycles. Let

$$P: v_0, e_1, v_1, e_2, \dots, e_k, v_k$$

be a longest directed path in D'. For every node v in D, assign to it a color $c(v)$ from $\{0,1,2,3, \dots, k\}$ where $c(v)$ is the length of a longest directed path in D' starting at v.

Consider now a directed u, v-path

$$Q: u = u_0, u_1, \dots, u_\ell = v$$

in D' with $\ell > 0$. We show $c(u) > c(v)$. Suppose $c(v) = i$ which means a longest directed path in D' starting at v has length i. Let

$$R: v = v_0, v_1, \dots, v_i$$

be such a directed path. If for any $s, 0 \leq s \leq \ell - 1$, and $t, 0 \leq t \leq i$ we have $u_s = v_t$, then QR would contain a directed cycle, which cannot happen. Hence, QR is a directed path of length $\ell + i$, implying the inequality $c(u) \geq \ell + i > i = c(v)$. Thus, in any directed path in D' of positive length, the starting and ending nodes have distinct colors.

Now, we show that if $uv \in A$, then $c(u) \neq c(v)$. If $uv \in A'$, then the directed path u, v is a directed path in D' and so $c(u) > c(v)$ from the previous paragraph. If $uv \notin A'$, then $D' \cup \{uv\}$ has a directed cycle C; this is from the fact that A' was as small as possible to allow D' not to have directed cycles. But C must contain uv as an arc since otherwise, D' has C as s directed cycle. But then $C - uv$ would be a directed v, u-path in D' and so $c(v) > c(u)$ from the previous paragraph. In either case, $c(u) \neq c(v)$.

It follows that c is a proper $(k + 1)$-coloring of D, and hence, of the underlying graph G. Hence,

$$k + 1 \geq \chi(G) \text{ or } k \geq \chi(G) - 1$$

as desired.

□

We now turn to connectedness which in digraphs is more subtle than that in undirected graphs. There are two types of connectedness to consider:

Definition. A digraph is **weakly connected** if its underlying graph is connected.

Definition. A digraph is **strongly connected** if for every two nodes v, w, there exists a directed v, w-walk.

Intuitively, "strong connectedness" means one can start at any node, follow a sequence of arcs oriented in the correct direction, and reach any node as a consequence.

In Figure V.A.3.b., we contrast a weakly connected vs. a strongly connected digraph. The only difference is that the orientation of the bottom arc of the triangle has been switched.

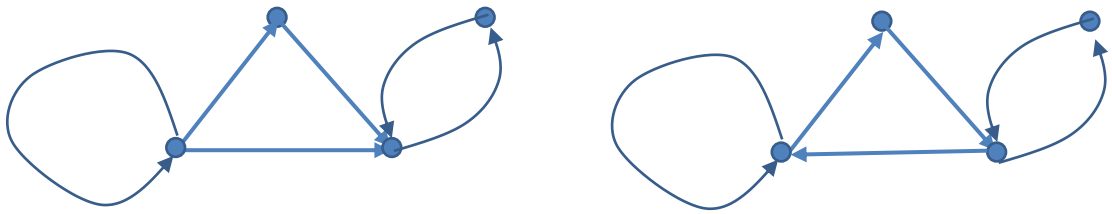


Figure V.A.3.b. A weakly connected and a strongly connected digraph with the same underlying graph.

The analog of a component for an undirected graph is called a directed component. We define this concept via equivalence classes under the equivalence relation $v \sim w$ iff there is a v, w-directed path and a w, v-directed path.

Definition. Let $N' = [u]$ be an equivalence class under $\sim$ above; the set N' is the set of all nodes x such that there is a u, x-directed path and an x, u-directed path. Then the induced subdigraph $D[N']$ is the **directed component** of D that contains u.

In Figure V.A.3.c., we show the directed components of the weakly connected graph on the left of Figure V.A.3.b.

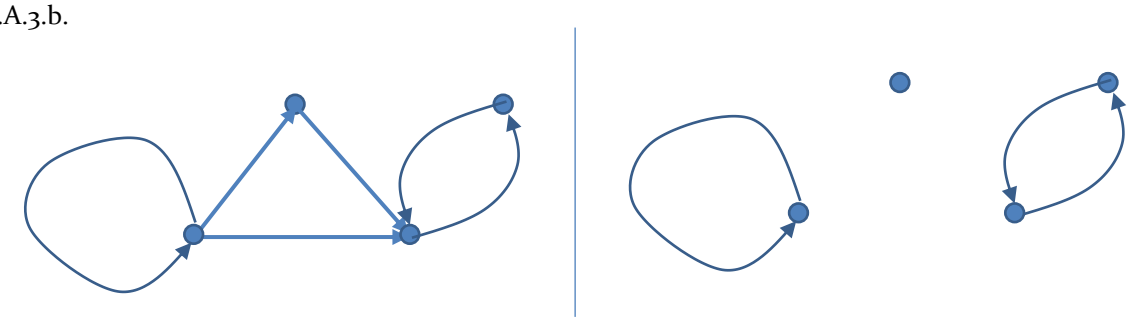


Figure V.A.3.c. A weakly connected digraph and its directed components.

V.B. FAMILIES OF DIGRAPHS

V.B.1. TOURNAMENTS

We now look at a specific family of digraphs called "tournaments."

Definition. A **tournament** is an orientation of K_n.

This name arises by considering a round-robin tournament with n participants where every participant plays every other participant. We assume no draws. The tournament is a digraph whose nodes are the participants. We draw an arc uv if participant u defeats participant v.

Since $\chi(K_n) = n$, we obtain the following corollary to Theorem V.A.3.a:

Corollary V.B.1.a. Every tournament has a **Hamilton directed path**, i.e. a directed path of length $n - 1$.

The first tournaments we consider are those without any directed cycles. Corollary V.A.4.a turns out to be a useful lemma in proving that there is essentially only one tournament on n nodes that has no directed cycles.

Theorem V.B.1.b. All tournaments on n nodes that have no directed cycles are isomorphic.

Proof. Let T be a tournament on n nodes with no directed cycles. Let

$$P: v_1, a_2, v_2, a_3, v_3, \dots, a_n, v_n$$

be a Hamilton directed path in T whose existence is guaranteed by Corollary V.A.4.a. If $a = v_i v_j$ is an arc such that $i > j$, then

$$C: v_j, a_{j+1}, v_{j+1}, \dots, v_i, a, v_j$$

is a directed cycle, which cannot happen. This implies every arc $v_i v_j$ of T is oriented such that $i < j$.

If T' is a tournament on n nodes with Hamilton directed path

$$Q: w_1, b_2, w_2, \dots, b_n, w_n$$

then every arc $w_i w_j$ of T' must also be oriented such that $i < j$. The mapping $\phi: N(T) \to N(T')$ given by $\phi(v_i) = w_i$ preserves the orientation of the arcs and, hence, is an isomorphism.

□

If T is a tournament with no directed cycles with Hamilton directed path

$$P: v_1, a_2, v_2, a_3, v_3, \dots, a_n, v_n,$$

and we have encoded the arcs where $a = v_i v_j$ means participant v_i defeated participant v_j, then we have a natural overall ranking of the participants: v_1 wins first place, v_2 wins second place, and so on.

On the other extreme from tournaments with no directed cycles, we consider tournaments that are strongly connected. We piece together a powerful result proved by Moon in 1966 that says every node is on a directed cycle of every possible length. We start with the following:

Lemma V.B.1.c. Every node of a strongly connected tournament with $n \geq 3$ nodes lies on a directed triangle.

Proof. Let T be a strongly connected tournament with $n \geq 3$ nodes and let u be a node of T. Since T is strongly connected, $d_{\text{in}}(u) \geq 1$ and $d_{\text{out}}(u) \geq 1$. Let $S_{\text{in}}(u)$ and $S_{\text{out}}(u)$ be the sets of all nodes that are in-neighbors and out-neighbors of u, respectively.

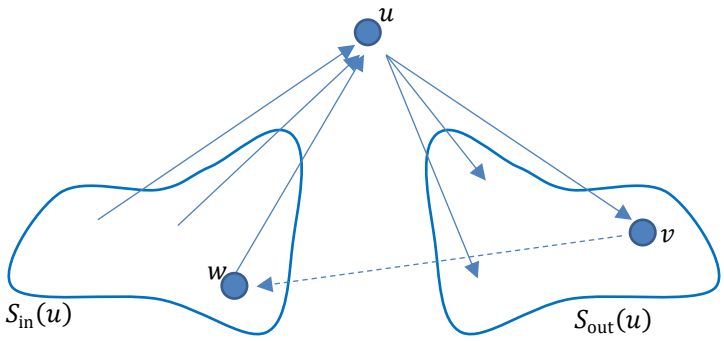


Figure V.B.1.d. The node u and the sets $S_{\text{in}}(u)$ and $S_{\text{out}}(u)$

If there is no arc from a node in $S_{\text{out}}(u)$ to one in $S_{\text{in}}(u)$, then there would be no directed path starting in $S_{\text{out}}(u)$ and ending in $S_{\text{in}}(u)$; hence, T would not be strongly connected. But since T is strongly connected, there must be an arc vw for some $v \in S_{\text{out}}(u)$ and some $w \in S_{\text{in}}(u)$. Hence, u, v, w, u is a directed triangle in T.

□

Lemma V.B.1.e. In every strongly connected tournament on n nodes, if $3 \leq k < n$ and u lies on a directed k-cycle, then u lies on a directed $(k+1)$-cycle.

Proof. Let T be a strongly connected tournament on n nodes, let k with $3 \leq k < n$ be given and let u lie on a directed k-cycle

$$C\colon\ u = v_0, a_1, v_1, a_2, v_2, \ldots, a_k, v_k = u.$$

Case 1. If there exists a node w not on C such that $v_p w$ and $w v_q$ are arcs for some distinct nodes v_p and v_q, then there must be some index i such that $v_i w$ and $w v_{i+1}$ are arcs. Then u is on the $(k+1)$-cycle

$$C'\colon u = v_0, a_1, v_1, a_2, v_2, \ldots, v_i, a, w, a', v_{i+1}, \ldots, a_k, v_k = u.$$

Case 2. Otherwise, for every node w not on C, either every arc joining C to w leaves C or every arc joining C to w enters C. Let $S_{\text{out}}(C)$ and $S_{\text{in}}(C)$ be the sets of nodes not on C that are out-neighbors and in-neighbors of nodes on C, respectively. These two sets do not intersect since otherwise, w would exist as in case 1. These two sets cannot both be empty as there are nodes not on C (this is a consequence of $k < n$). If $S_{\text{out}}(u)$ is empty, then there is no way to enter $S_{\text{in}}(u)$ from C and so T is not strongly connected. Similarly, if $S_{\text{in}}(u)$ is empty, then there is no way to leave $S_{\text{out}}(u)$ into C and so T is not strongly connected. It follows that both sets must be nonempty. We are in the situation depicted in Figure V.B.1.f.

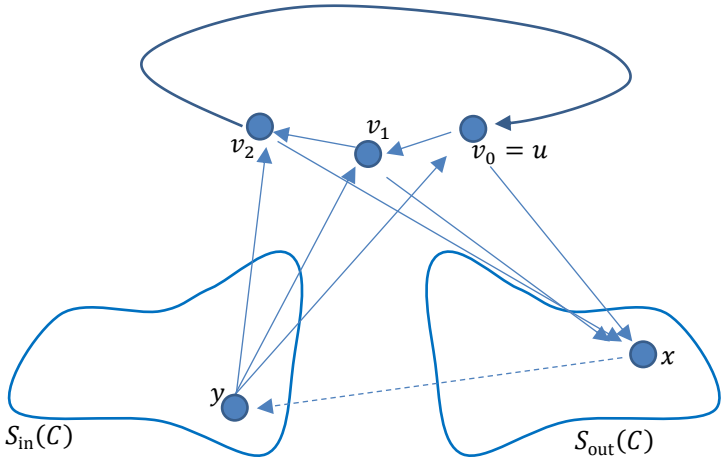


Figure V.B.1.f.

Since T is strongly connected, there must be a node $x \in S_{\text{out}}(C)$ and a node $y \in S_{\text{in}}(C)$ such that xy is an arc. Then

$$C' : u = v_0, x, y, v_2, \dots, v_k = u$$

is a directed $(k+1)$-cycle that contains u.

□

We now have Moon's theorem by induction on k, using Lemma V.B.1.c as the base case and applying Lemma V.B.1.e for the inductive step.

Theorem V.B.1.g. (Moon, 1966) In any strongly connected tournament with $n \geq 3$ nodes and any k with $3 \leq k \leq n$, every node is on a directed k-cycle.

Corollary V.B.1.h. If T has a directed Hamiltonian cycle, then Moon's theorem applies to T.

V.B.2. PARTIALLY ORDERED SETS (POSETS)

Definition. A **partial order** on a set S is a relation $\sim$ such that

- For all $a \in S$, $a \sim a$;
- For all $a, b \in S$, if $a \sim b$ and $b \sim a$, then $a = b$.
- For all $a, b, c \in S$, if $a \sim b$ and $b \sim c$, then $a \sim c$.

In short, a partial order is a relation that is reflexive, anti-symmetric, and transitive.

Anti-symmetry is equivalent to saying at most one of $a \sim b$ or $b \sim a$ holds.

Definition. We note that if $a \neq b$, then there is no guarantee that $a \sim b$ or $b \sim a$. Two elements a, b are **comparable** if $a \sim b$ or $b \sim a$; they are **incomparable** if neither statement holds.

Definition. A **partially ordered set** (or **poset**) is a set S equipped with a partial order $\leq$.

We can model posets as directed graphs in the following manner: Given a poset $S, \leq$ we define the directed graph D whose node set is S and where uv is an arc iff $u \sim v$ and $u \neq v$.

Definition. A digraph that models a poset is called a **poset graph**.

Observations. In a poset graph,

- There are no loops.
- There is at most one arc joining any two given nodes.
- If $u \neq v$ and there is a directed u, v-path, then uv is an arc.

Definition. The sub-digraph induced by a directed u, v-path is called a **chain digraph**. This captures all of the order statements implied by

$$v \sim v_1 \sim v_2 \sim \ldots \sim v_{k-1} \sim u.$$

Definition. If $S' \subseteq S$ has the property that every two elements of S' are comparable, then S' is called a **chain**.

Definition. If S itself is a chain, then $\sim$ is a **total order**.

Definition. If $S' \subseteq S$ has the property that every two distinct elements of S' are incomparable, then S' is called an **antichain**. We observe that a sub-digraph induced by an antichain has no arcs.

Theorem V.B.2.a. If D is a poset graph, then D has no directed cycles.

Proof. Let C be a directed cycle in D. Suppose u, v are arbitrarily chosen nodes on C. Then there exists a directed u, v-path and a directed v, u-path. By transitivity, uv and vu are both arcs and so $u \sim v$ and $v \sim u$. This implies $u = v$ and so every node on C is equal. Hence, C is a loop. Since D has no loops, C does not exist.

¤

We also have the following result as a partial converse:

Theorem V.B.2.b. Let $D = (N, A)$ be a digraph without directed cycles. Let $\sim$ be the relation on N where $u \sim v$ if there exists a u, v-directed path. Then N is a poset with partial order $\sim$.

Proof. We check the conditions on $\sim$. For any node v, the trivial v, v-directed path exists and so $v \sim v$. Suppose $u \sim v$ and $v \sim u$. Then there exist u, v- and v, u-directed paths. If $u \neq v$, then concatenating these directed paths results in a nontrivial directed closed directed walk. The portion of this walk between the closest occurrences of a repeated node form a directed cycle. This cannot happen, so $u = v$. Finally, if $u \sim v$ and $v \sim w$, then there are w, v- and v, u-directed paths. Concatenating these forms a w, u-directed walk and so w, u-directed paths exist (the shortest w, u-directed walk is a directed path.) Thus, $u \sim w$ and so $\sim$ is a partial order and N is a poset armed with the partial order $\sim$.

▫

Definition. If $D = (N, A)$ is a digraph, then its **transitive closure** $\widetilde{D}$ has the same node set N and for nodes u and w, if uw is an arc of D or there is a node v such that uv and vw are arcs of D, then uw is an arc of $\widetilde{D}$. A digraph D is **transitively closed** if $D = \widetilde{D}$.

Theorem V.B.2.c. A digraph is a poset graph iff it is transitively closed digraph and it has no directed cycles.

Now we show how the directed components of a digraph can be used to construct a poset graph.

Let D be any digraph and let $C_1, C_2, \ldots, C_k$ be its directed components. We construct a digraph D' whose node set is $N' = \{C_1, C_2, \ldots, C_k\}$ and where for $i \neq j$, an arc is drawn from C_i to C_j iff there is a u, v-directed path for some $u \in C_i$ and $v \in C_j$.

Theorem V.B.2.d. The digraph D' constructed above is a poset graph.

Proof. We consider the relation $\sim$ among the directed components C_i given by $C_i \sim C_j$ iff there is an arc drawn from C_i to C_j in D' or if $i = j$. By definition, $C_i \sim C_i$ for all i. Next, suppose $C_i \sim C_j$ and $C_j \sim C_i$. We suppose $i \neq j$ (since otherwise, we obtain anti-symmetry at once.) Then there are arcs drawn from C_i to C_j and from C_j to C_i. This implies there are $u, u' \in C_i$ and $v, v' \in C_j$ such that there exist u, v- and v', u'-directed paths P, P' in D. We also note that there exists a u', u-directed path R in D as well. Let w be any node in C_j. Then since $v \in C_j$, there exists a v, w-directed path Q in D. Hence, the concatenation PQ is a u, w-directed walk in D. Also, there exists a w, v-directed path Q' in D, implying the concatenation $Q'P'R$ is a w, u-directed walk in D. It follows that u and w are in the same directed component and so $C_i = C_j$. Finally, suppose $C_i \sim C_j$ and $C_j \sim C_k$. If $i = j$ or $j = k$, then we have $C_i \sim C_k$ immediately. Otherwise, there exist nodes $u \in C_i, v \in C_j, v' \in C_j, w \in C_k$ such that there are u, v-, v, v'-, and v', w-directed paths in D. Then the concatenation of these is a u, w-directed walk in D, implying that there exists a u, w-directed path in D and so $C_i \sim C_k$ follows. Hence, $\sim$ is reflexive, anti-symmetric, and transitive and so $N, \sim$ is a poset. Hence, D' is a poset graph.

▫

We observe that in the original digraph D, all arcs between C_i and C_j must point in the same direction. Otherwise, C_i and C_j are the same directed component.

Corollary V.B.2.e. If T is a tournament, then T' is a chain graph.

Proof. We already know T' is a poset graph. Also, for any $i \neq j$, let $u \in C_i$ and $v \in C_j$ be given. Then either uv or vu is an arc in T, implying C_iC_j or C_jC_i is an arc in T'. Hence, N' is a chain and T' is a chain graph.

□

The upshot of this is that the directed components of a tournament can be put into order,

$$C_1 > C_2 > C_3 > \cdots > C_k$$

which provides a preliminary scheme for ranking the participants in the tournament: Those who are in component C_1 place higher than those in component C_2, and so on.

V.C. WEIGHTED DIGRAPHS

V.C.1. WEIGHT FUNCTIONS AND ARC DIRECTIONS

Many applications of digraphs involve assigning weights to the arcs.

Definition. Given a digraph $D = (N, A)$, a **weight function** is a function $w: A \to \mathbb{R}$. The specific value $w(a)$ is the **weight** assigned to arc a. The same terminology is used for functions $w: N \to \mathbb{R}$.

Often, restrictions are placed on the weights such as requiring the weights be nonnegative or that they be integers.

Often, weights are used to denote some sort of cost, distance, or effort to traverse a given arc. In this interpretation, weight is viewed with a negative connotation: A lower weight is preferred to a higher one. Also, infinite weight is allowed in some applications, where the cost or effort is prohibitive. Usually, this is in applications where one wishes to find structures of minimum weight and infinite weight is assigned to arcs that are not present in D, but that are then added so that every ordered pair of distinct nodes is assigned an arc.

Problems involving networks (Section V.C.3) involve additional terminology. For instance, **capacity** and **flow** are commonly used "benevolent" weight functions, where a higher quantity is preferred to a lower one.

Definition. If

$$Q\colon\ v_0, a_1, v_1, \dots, a_k, v_k$$

is a directed trail (no repeating arcs) in D, then

- The **total weight** of Q is the quantity $\sum_{i=1}^{k} w(a_i)$,
- The **minimum weight** of Q is $\min\{w(a_1), w(a_2), \dots, w(a_k)\}$
- The **maximum weight** of Q is $\max\{w(a_1), w(a_2), \dots, w(a_k)\}$

In most applications Q is a directed path, but there are situations where these concepts are applied to directed cycles.

Dijkstra's algorithm (Section V.C.2) is commonly used to find directed paths having minimum weight among all directed paths from one given node to another.

In networks, undirected paths and cycles play a role. Here, we will want to distinguish whether an arc proceeds "forward" or "backward" along such a path.

Definition. Given a digraph D with underlying graph G, if

$$Q\colon v_0, e_1, v_1, \dots, e_k, v_k$$

is a path or cycle in G then an arc $a_i = v_{i-1}v_i$ is called a **forward arc** along Q and an arc $a_i = v_i v_{i-1}$ is called a **backward arc** along Q.

V.C.2. DIJKSTRA'S ALGORITHM

In 1956, Dijkstra formulated one of the earliest digraph-based algorithms. This algorithm is designed to solve the following problem (we note that the weight function w takes on only positive values):

Problem. Let $D = (N, A)$ be a digraph equipped with a weight function $w: A \to \mathbb{R}^+$. Also, let s be a distinguished node. For all nodes u, find the minimum total weight among all s,u-directed paths. If no s,u-directed path exists, return ∞.

Intuitively, Dijkstra's algorithm keeps track of tentative minimum total weights of directed paths from s to every node u in a function $f(u)$. We start by declaring $f(s) = 0$ and $f(u) = \infty$ for all nodes $u \neq s$. Each node begins as an "unmarked" node and a node becomes "marked" after we are done measuring distances using that node. An intuitive list of steps of Dijkstra's algorithm follows:

Steps:

1. If all unmarked nodes z satisfy $f(z) = \infty$, then there are no s,z-directed paths; the algorithm stops.
2. If $f(z) < \infty$ for some unmarked node z, find a closest unmarked node x to s, i.e., an unmarked node where $f(x)$ is minimized.
3. For every arc of the form xy where y is unmarked, redefine $f(y)$ to be the smaller of $f(y)$'s current value and the quantity $w(xy) + f(x)$.
4. Mark node x and return to step 1.

Observation. When a node x becomes marked, every node u such that $f(u) < f(x)$ must be marked.

More formally, Dijkstra's solution boils down to the following algorithm:

Dijkstra's Algorithm.

Inputs: A digraph $D = (N, A)$, a weight function $w: A \to \mathbb{R}^+$, and a distinguished node s.

Initialization: Define two functions $f: N \to [0, \infty) \cup \{\infty\}$ and $g: N \to \{0,1\}$ and initialize these as follows:

$$f(s) = 0; f(u) = \infty \text{ for } u \neq s$$
$$g(u) = 0 \text{ for all } u \in N$$

The function $f(u)$ represents a tentative minimum weight among s,u-directed paths.

The function $g(u)$ represents whether the value of $f(u)$ is fixed for that node u, i.e., if the node u has been completely analyzed; if $g(u) = 0$, then $f(u)$ may change whereas if $g(u) = 1$, then $f(u)$ is fixed at its current value.

Also, let x be a variable standing for a node; initialize $x = s$.

Steps:

1. Consider the set $Y = \{f(y) \text{ such that } g(y) == 0\}$
2. If $\min(Y) = \infty$, return the function f and exit
3. If $\min(Y) < \infty$, find a node y such that $f(y) = \min(Y)$ and replace $x := y$
4. For the node x,

   a. For each arc xy, if $g(y) == 0$ replace $f(y) \coloneqq \min\{f(y), f(x) + w(xy)\}$
   b. Set $g(x) = 1$
5. Go to step 1

For notational convenience, let $\delta(u)$ be the minimum total weight among all s,u-directed paths.

Theorem V.C.2.a. The function $f: V \to \mathbb{R}$ returned by Dijkstra's algorithm provides the minimum total weights among all s,u-directed paths.

Proof. Suppose the vertices are marked in the order $v_0 = s, v_1, v_2, \ldots, v_k$. We claim for all $i, 0 \leq i \leq k$, that $f(v_i) = \delta(v_i)$. We proceed by induction on i, noting that $f(v_0) = 0 = \delta(s) = \delta(v_0)$ from the initialization and from the fact that $f(v_0)$ never changes.

Now, suppose $f(v_j) = \delta(v_j)$ for all j, $0 \leq j \leq i$ and we consider v_{i+1} after when $f(v_{i+1})$ had been updated for the last time, but just before v_{i+1} is marked. Since the value of $f(v_{i+1})$ is the sum of weights of arcs along some s, v_{i+1}-directed path, we must have $\delta(v_{i+1}) \leq f(v_{i+1})$. Let Q be an s, v_{i+1}-directed path of minimum weight. Since s is marked but v_{i+1} is unmarked, there must be a node x along Q such that every node along Q up to x is marked but the next node y is unmarked. Let Q' be the s,x-directed subpath of Q. Since Q has minimum total weight, Q' must also have minimum total weight $\delta(x)$, being a subpath. Since x is marked, $f(x) = \delta(x)$. Since x is marked and y is adjacent to x, the value $f(y)$ was updated when x was marked, hence

$$f(y) \leq f(x) + w(xy).$$

Also, $f(x) + w(xy) = \delta(x) + w(xy) \leq \delta(v_{i+1})$. Since y is unmarked when v_{i+1} was to be marked, $f(v_{i+1}) \leq f(y)$. We now have

$$f(v_{i+1}) \leq f(y) \leq f(x) + w(xy) \leq \delta(v_{i+1}) \leq f(v_{i+1})$$

which forces $\delta(v_{i+1}) = f(v_{i+1})$. This completes the inductive step and the proof, observing that for all nodes z left unmarked, $f(z)$ was never reduced from $f(z) = \infty$.

□

V.D. NETWORK FLOWS

V.D.1. DEFINITIONS

Definition. A **network** is a digraph $D = (N, A)$. To facilitate the exposition, we do not allow for loops or parallel arcs; we assume all parallel arcs are combined into a single arc. We do not exclude the possibility of anti-parallel arcs, however; in fact, a common practice is to introduce anti-parallel arcs to aid in analyzing networks. We will see this when introducing residual flows.

Definition. A **capacity function** is a function $c: A \to [0, \infty]$.

Each of the following is defined for a given network $D = (N, A)$ and capacity function c.

Definitions.

- A **flow** is a function $f: A \to [0, \infty)$ such that $f(a) \leq c(a)$ for all arcs $a \in A$.
- Given a node $n \in N$ and a flow f on D, the **total in-flow** and **total out-flow** on n are

$$f_{\text{in}}(n) = \sum_{a=un} f(a)\,, f_{\text{out}}(n) = \sum_{a=nw} f(a).$$

  These can be visualized as the total quantity of material flowing into node n and flowing out of node n, respectively.

- The **net flow** on n is defined as $f_{\text{net}}(n) = f_{\text{out}}(n) - f_{\text{in}}(n)$.
- A node n is **balanced** if $f_{\text{net}}(n) = 0$, **producing** if $f_{\text{net}}(n) > 0$, and **consuming** if $f_{\text{net}}(n) < 0$.
- A flow is a **circulation** if every node is balanced.
- Often, two disjoint sets S and T of nodes are designated where $s \in S$ is a **source** node and $t \in T$ is a **sink** node. Usually, it is required that every producing node be a source and every consuming node be a sink. Also, often there is only one source node and one sink node.

Theorem V.D.1.a. For any flow f, the total in-flow and total out-flow always equal.

Proof. For every node $p \in N$, if $pn \notin A$, then we extend f by defining $f(pn) = 0$. We have

$$\begin{aligned}\sum_{n\in N} f_{\text{in}}(n) &= \sum_{n\in N}\sum_{a=pn} f(a)\\ &= \sum_{n\in N}\sum_{p\in N} f(pn)\\ &= \sum_{p\in N}\sum_{n\in N} f(pn)\\ &= \sum_{p\in N}\sum_{a=pn} f(pn)\\ &= \sum_{p\in N} f_{\text{out}}(p)\end{aligned}$$

Thus, the total in-flow must always equal the total out-flow.

▫

This result can be visualized more succinctly by noting that the flow on every arc contributes the same amount to the total in-flow and the total-outflow.

Corollary V.D.1.b. The total net flow must be zero, i.e.,

$$\sum_{n\in N} f_{\text{net}}(n) = \sum_{n\in N} \big(f_{\text{out}}(n) - f_{\text{in}}(n)\big) = \sum_{n\in N} f_{\text{out}}(n) - \sum_{n\in N} f_{\text{in}}(n) = 0.$$

Corollary V.D.1.c. If every producing node is in S and every consuming node is in T, then

$$\sum_{s\in S} f_{\text{net}}(s) + \sum_{t\in T} f_{\text{net}}(t) = 0.$$

Definition. If every producing node is in S and every consuming node is in T, then the **value** of a flow f is

$$v(f) = \sum_{s\in S} f_{\text{net}}(s)\,.$$

In many applications, the goal is to find a flow that maximizes $v(f)$.

Theorem V.D.1.d. Let $D = (N, A)$ be a network and let f, f^* be flows. If $C: n_0, a_1, n_1, a_2, \dots, a_k, n_k = n_0$ is a directed cycle, with

$$f^*(a) = \begin{cases} f(a) + m, \text{ if } a = a_i \text{ on } C \\ f(a), \quad \text{if } a \text{ is not on } C \end{cases}$$

then for all nodes n, $f^*_{\text{net}}(n) = f_{\text{net}}(n)$ and $v(f^*) = v(f)$.

Proof. We note that for a node n not on C and any arc a incident with n, $f^*(a) = f(a)$ and so $f^*_{\text{net}}(n) = f_{\text{net}}(n)$. For n_i on C, we have $f^*(a_i) = f(a_i) + m$ and $f^*(a_{i+1}) = f(a_{i+1}) + m$. For all other arcs a incident with n_i, we have $f^*(a) = f(a)$. We then have $f^*_{\text{in}}(n_i) = f_{\text{in}}(n_i) + m$ and $f^*_{\text{out}}(n_i) = f_{\text{out}}(n_i) + m$, implying $f^*_{\text{net}}(n_i) = f_{\text{net}}(n_i)$.

Since the net flows are unchanged, the value of the flow remains unchanged as well.

▫

Corollary V.D.1.e. If a and a' are anti-parallel, then reducing the flow by $\min\{f(a), f(a')\}$ has no effect on the net flows or the value of the flow while reducing the flow on one of the arcs to zero.

Given corollary V.D.1.e., we may assume the flow is zero on at least one arc of an anti-parallel pair of arcs.

V.D.2. AUGMENTING PATHS

Let $D = (N, A)$ be a network with capacity function c and flow f. Also, suppose a source node s and a sink node t are specified with $S = \{s\}, T = \{t\}$.

Definitions.

Given an *undirected* s,t-path

$$P\colon\ s = n_0, a_1, n_1, \dots, a_k, n_k = t$$

where $a_i = n_{i-1}n_i$ or $a_i = n_i n_{i-1}$,

- we say a_i is a **forward arc** if $a_i = n_{i-1}n_i$, and a_i is a **backward arc** if $a_i = n_i n_{i-1}$.
- If for every forward arc a, $f(a) < c(a)$ and for every backward arc, $0 < f(a)$, we say P is an **augmenting path.**

We note that $n_1, n_2, \dots, n_{k-1}$ are all balanced nodes because only s can be producing and only t can be consuming.

Definition. Let $P\colon\ s = n_0, a_1, n_1, \dots, a_k, n_k = t$ be an undirected s,t-path, $F = \{a_i : a_i \text{ is a forward arc on } P\}$ and $B = \{a_j : a_j \text{ is a backward arc on } P\}$. Then the quantity

$$\delta(P) = \min(\{c(a) - f(a) : a \in F\} \cup \{f(a)\colon\ a \in B\}).$$

is the **augmenting capacity** of P.

Theorem V.D.2.a. Given $D = (N, A), c, f$, and P as above, we define a function $f^*\colon A \to \mathbb{R}$ given by

$$f^*(a) = \begin{cases} f(a)\colon & a \text{ is not on } P \\ f(a_i) + \delta(P)\colon & a = a_i \in F \\ f(a_j) - \delta(P)\colon & a = a_j \in B \end{cases}$$

Then f^* is a flow on D with capacity function c. Furthermore, every node $n_i, 1 \le i \le k-1$, remains balanced under f^* and $v(f^*) = v(f) + \delta(P)$.

Proof. To show f^* is a flow, we must verify that for all $a \in A$, $0 \le f^*(a) \le c(a)$.

If a is not on P, then $f^*(a) = f(a)$ and we already know $0 \le f(a) \le c(a)$ because f is a flow.

If $a = a_i \in F$, then

$$0 \le f(a_i) \le f(a) = f^*(a_i) = f(a_i) + \delta(P) \le f(a_i) + c(a_i) - f(a_i) = c(a_i) = c(a).$$

If $a = a_j \in B$, then

$$0 \le f(a_j) - \min\{f(a)\colon\ a \in B\} \le f^*(a) = f^*(a_j) = f(a_j) - \delta(P) \le f(a_j) \le c(a_j) = c(a).$$

To show every balanced node remains balanced, let $n \in N$ be a balanced node. If n is not on P, then every arc incident with n is not on P. Thus, for all arcs a incident with n, $f^*(a) = f(a)$ and so

$$0 = f_{\text{net}}(n) = \sum_{a=nu} f(a) - \sum_{a=un} f(a) = \sum_{a=nu} f^*(a) - \sum_{a=un} f^*(a) = f^*_{\text{net}}(n)$$

and so n remains balanced.

If $n = n_i, 1 \le i \le k-1$ on P, there are four cases to consider, depending on whether arcs a_i and a_{i+1} are forward or backward. These two arcs are the only ones whose changes can affect $f^*_{\text{net}}(n)$.

Case 1. Both a_i and a_{i+1} are forward. Then $f^*_{\text{in}}(n) = f_{\text{in}}(n) + \delta(P)$ from arc a_i and $f^*_{\text{out}}(n) = f^*_{\text{out}} + \delta(P)$ from arc a_{i+1}. Hence, $f^*_{\text{net}}(n) = f_{\text{net}}(n) - \delta(P) + \delta(P) = f_{\text{net}}(n) = 0$.

Case 2. Arc a_i is forward and arc a_{i+1} is backward. Then $f^*_{\text{in}}(n) = f_{\text{in}}(n) + \delta(P)$ from arc a_i and $f^*_{\text{in}}(n) = f^*_{\text{in}} - \delta(P)$ from arc a_{i+1}. Hence, $f^*_{\text{net}}(n) = f_{\text{net}}(n) - \delta(P) + \delta(P) = f_{\text{net}}(n) = 0$.

Case 3. Arc a_i is backward and arc a_{i+1} is forward. Then $f^*_{\text{out}}(n) = f_{\text{out}}(n) - \delta(P)$ from arc a_i and $f^*_{\text{out}}(n) = f^*_{\text{out}} + \delta(P)$ from arc a_{i+1}. Hence, $f^*_{\text{net}}(n) = f_{\text{net}}(n) - \delta(P) + \delta(P) = f_{\text{net}}(n) = 0$.

Case 4. Both a_i and a_{i+1} are backward. Then $f^*_{\text{out}}(n) = f_{\text{out}}(n) - \delta(P)$ from arc a_i and $f^*_{\text{in}}(n) = f^*_{\text{in}} - \delta(P)$ from arc a_{i+1}. Hence, $f^*_{\text{net}}(n) = f_{\text{net}}(n) - \delta(P) + \delta(P) = f_{\text{net}}(n) = 0$.

Finally, we examine $v(f^*)$. The only arc incident with s whose flow value is adjusted is arc a_1. If arc a_1 is forward, then $f^*_{\text{out}}(s) = f_{\text{out}}(s) + \delta(P)$ and $f^*_{\text{in}}(s) = f_{\text{in}}(s)$; hence, $v(f^*) = f^*_{\text{net}}(s) = f_{\text{net}}(s) + \delta(P)$. If arc a_1 is backward, then $f^*_{\text{out}}(s) = f_{\text{out}}(s)$ and $f^*_{\text{in}}(s) = f_{\text{in}}(s) - \delta(P)$; hence, $v(f^*) = f^*_{\text{net}}(s) = f_{\text{net}}(s) + \delta(P)$.

□

Intuitively, if there exists an augmenting path P, then we can increase the flow by $\delta(P)$ on forward arcs along P and decrease the flow by $\delta(P)$ on backward arcs along P with the result being a new flow whose value is increased by $\delta(P)$. This is the underlying impetus behind the Ford-Fulkerson algorithm described in Section V.D.3.

V.D.3. THE FORD-FULKERSON ALGORITHM

The Ford-Fulkerson algorithm is intended to find a flow f on a network $D = (N, A, c)$ with the maximum value $v(f)$. Armed with the result in Theorem V.D.2.a, the algorithm is quite brief:

Inputs: A network $D = (N, A)$ with a capacity function $c: A \to [0, \infty]$. Also, a source node s and a sink node t.

Output: A flow f whose value $v(f)$ is maximum among all flows for D and c.

Initialization: Declare $f(a) = 0$ for all arcs $a \in A$.

Steps.

1. Determine if an augmenting s,t-path exists for f.
2. If there is no augmenting s,t-path for f, then return f.
3. Choose an augmenting s,t-path P for f.
4. Define F as the set of forward arcs along P and B as the set of backward arcs along P.
5. Define $\delta(P) = \min(\{c(a) - f(a): a \in F\} \cup \{f(a):\ a \in B\})$.
6. Redefine the function $f: A \to \mathbb{R}$ by
$$f(a) := \begin{cases} f(a): & a \text{ is not on } P \\ f(a_i) + \delta(P): & a = a_i \in F \\ f(a_j) - \delta(P): & a = a_j \in B \end{cases}$$
7. Return to step 1.

There are several things to consider at this point:

- First, the details of how to determine whether an augmenting s,t-path exists are left unspecified. For this, one could perform a depth-first or breadth-first search rooted at s, following forward arcs where $c(a) - f(a) > 0$ or backward arcs where $f(a) > 0$.
- Second, it is theoretically possible that this process never terminates if irrational capacities are allowed. In practice, one allows only integer capacities and it can be shown that the algorithm must terminate in this case.
- Third, even if capacities are integers, there are networks where implementations of this algorithm can perform very inefficiently.

VI. LINEAR ALGEBRA APPLIED TO GRAPH THEORY

VI.A. REVIEW OF FUNDAMENTAL CONCEPTS IN LINEAR ALGEBRA

VI.A.1. VECTOR SPACES

If you ask a mathematician what a vector is, you might get the unhelpful answer, "an element of a vector space." This is perhaps he or she is so used to this concept that it has become second nature. We review what vector spaces are and then offer numerous examples. For our purposes, a **scalar** is a real number, though one can consider vector spaces using different sets of scalars. We will denote scalars with lower case italic and non-bold variables such as a, b, c and vectors with lower case non-italic and bold variables such as $\mathbf{u}, \mathbf{v}, \mathbf{w}$.

Definition. A **vector space** over $\mathbb{R}$ is a nonempty set V together with two operations, vector addition (the sum of two vectors $\mathbf{u}, \mathbf{v} \in V$ is denoted $\mathbf{u} + \mathbf{v}$) and scalar multiplication (if c is a real scalar and $\mathbf{v} \in V$ is a vector, then $c\mathbf{v}$ is the **scalar product** of c and $\mathbf{v}$) where the following properties hold for all vectors and scalars:

- Existence of zero vector: There is a vector $\mathbf{0} \in V$, called the **zero vector.**
- Vector addition …
  - … is closed in V: For all $\mathbf{u}, \mathbf{v} \in V$, we have $\mathbf{u} + \mathbf{v} \in V$.
  - … is commutative: For all vectors $\mathbf{u}, \mathbf{v} \in V$, we have $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.
  - … is associative: For all vectors $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$, we have $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.
  - … is invertible: For all $\mathbf{u} \in V$, there exists a vector $-\mathbf{u} \in V$ such that $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.
  - … has $\mathbf{0}$ as its identity: For all $\mathbf{u} \in V$, we have $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$.
- Scalar multiplication …
  - … is closed in V: For all $\mathbf{u} \in V$ and $c \in \mathbb{R}$, we have $c\mathbf{u} \in V$.
  - … is associative: For all $\mathbf{u} \in V$ and $c, d \in \mathbb{R}$, we have $c(d\mathbf{u}) = (cd)\mathbf{u}$.
  - … distributes over vector and scalar addition: For all $\mathbf{u}, \mathbf{v} \in V$ and $c, d \in \mathbb{R}$, we have $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$ and $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.
  - … has 0 as its zero: For all $\mathbf{u} \in V$, we have $0\mathbf{u} = \mathbf{0}$.
  - … has 1 as its unity: For all $\mathbf{u} \in V$, we have $1\mathbf{u} = \mathbf{u}$.

This is all very abstract. We offer several examples of vector spaces, the first of which should be familiar to those with some previous exposure to linear algebra.

Theorem VI.A.1.a. If $a\mathbf{u} = \mathbf{0}$, then $a = 0$ or $\mathbf{u} = \mathbf{0}$.

Proof. Suppose $a\mathbf{u} = \mathbf{0}$ and $a \neq 0$. Then

$$\mathbf{u} = 1\mathbf{u} = \left(\frac{1}{a}a\right)\mathbf{u} = \frac{1}{a}(a\mathbf{u}) = \frac{1}{a}\mathbf{0} = \mathbf{0}.$$

□

Corollary VI.A.1.b. If $a \neq 0$ and $\mathbf{u} \neq \mathbf{0}$, then $a\mathbf{u} \neq \mathbf{0}$.

This is the contrapositive of theorem VI.A.1.a.

Example VI.A.1.c. For a positive integer n, the vector space $\mathbb{R}^n$ is the set

$$\left\{ \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} : \text{ for all } i, 1 \leq i \leq n, r_i \in \mathbb{R} \right\}.$$

It has become canonical to write vectors in $\mathbb{R}^n$ as "column vectors."

Vector addition and scalar multiplication are defined by

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}; \qquad \mathbf{u} + \mathbf{v} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{bmatrix}, \qquad c\mathbf{u} = \begin{bmatrix} cu_1 \\ cu_2 \\ \vdots \\ cu_n \end{bmatrix}.$$

Given a vector $\mathbf{u} \in \mathbb{R}^n$, the entry u_i in the i^{th} position is called the i^{th} **coordinate of u**.

Example VI.A.1.d. Given a set S, the set of functions $f: S \to \mathbb{R}$ can be treated as a vector space V where vector addition and scalar multiplication are defined by

$$(f + g)(s) = f(s) + g(s); \;\; (cf)(s) = c \cdot f(s).$$

In this context, we are treating $f + g$ and cf as elements of V, while we are treating $f(s) + g(s)$ and $c \cdot f(s)$ as elements of $\mathbb{R}$. Thus, $(f + g)(s)$ should be interpreted as applying the function $f + g$ to the element $s \in S$ whereas $f(s) + g(s)$ should be interpreted as the result of adding the two real numbers $f(s)$ and $g(s)$. Similarly, $(cf)(s)$ is interpreted as the result of applying the function cf to the element s and $c \cdot f(s)$ is the result of multiplying c by the real number $f(s)$.

An alternative notation uses subscripts—f_s in place of $f(s)$—to copy the indexing notation for $\mathbb{R}^n$. Here, we are treating the set S as an index set.

VI.A.2. LINEAR COMBINATIONS AND SPAN

Definition. Given a finite set of vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p\}$, a **linear combination** of these vectors is a vector w written as the sum of the form

$$\mathbf{w} = \alpha_1 \mathbf{w}_1 + \alpha_2 \mathbf{w}_2 + \cdots + \alpha_p \mathbf{w}_p = \sum_{i=1}^{p} \alpha_i \mathbf{w}_i$$

where the α_i are real scalars.

Observation. The zero vector is a linear combination of any finite set of vectors:

$$\mathbf{0} = 0\mathbf{v}_1 + 0\mathbf{v}_2 + \cdots + 0\mathbf{v}_p.$$

Conventionally, this holds for the empty set of vectors as well; it is understood that the zero vector is the only linear combination of the empty set of vectors. Such a linear combination is sometimes called an "empty sum."

Definition. Given a set W of vectors (not necessarily finite) their **span** is the set of all linear combinations of vectors in W:

$$\text{span } W = \left\{ \sum_{i=1}^{p} \alpha_i \mathbf{w}_i : \ p \in \{0,1,2,\dots\}, \alpha_i \in \mathbb{R}, \mathbf{w}_i \in W \right\}.$$

We will say that W **generates** span W.

Observations.

- The zero vector is in span W since we allow $p = 0$, leading to the empty sum of vectors in W. If W is nonempty, we can take p to be positive and all of the α_i to be 0.
- If $\mathbf{w} \in W$, then $\mathbf{w} \in$ span W. This is immediate from $\mathbf{w} = 1\mathbf{w}$ being a linear combination with $p = 1$.
- More generally from the $p = 1$ case, if $\mathbf{w} \in W$, then for every scalar α, $\alpha\mathbf{w} \in$ span W. If $\mathbf{w} \neq \mathbf{0}$, then the set $\{\alpha\mathbf{w}: \alpha \in \mathbb{R}\}$ is the **line containing w.** One can show that the line containing $\mathbf{w}$ is also span $\{\mathbf{w}\}$.

Theorem VI.A.2.a. If $W = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p\}$, a finite set, then

$$\text{span } W = \left\{ \sum_{i=1}^{p} \alpha_i \mathbf{w}_i : \alpha_i \in \mathbb{R} \right\}.$$

Proof. Since every element in the set on the right—call this set X—is a linear combination of vectors in W, it is immediate that $X \subseteq$ span W. For the other inclusion, let $\mathbf{w} \in$ span W be given. Then

$$\mathbf{w} = \sum_{\substack{i_k \in \{1,2,3,\dots,p\} \\ 1 \le k \le q}} \alpha_{i_k} \mathbf{w}_{i_k} = \sum_{j=1}^{p} \sum_{\substack{i_k = j \\ 1 \le k \le q}} \alpha_{i_k} \mathbf{w}_{i_k} = \sum_{j=1}^{p} \left(\sum_{\substack{i_k = j \\ 1 \le k \le q}} \alpha_{i_k} \right) \mathbf{w}_j \in X.$$

One can show that for any set $W \subseteq V$, span W satisfies the properties of being a vector space.

Definition. Given a vector space V, we say that a subset $X \subseteq V$ is a **subspace** of V if X satisfies the properties of being a vector space. All closure properties are with respect to X.

With this in mind, if $W \subseteq V$, then span W is a subspace of V.

VI.A.3. LINEAR INDEPENDENCE

Linear independence is one of the central ideas in linear algebra.

Definition. We say that the vectors in a set W are **linearly independent** if for any finite set of vectors

$$\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\} \subseteq W,$$

the only way to express $\mathbf{0}$ as a linear combination of the w_i is

$$\mathbf{0} = 0\mathbf{w}_1 + 0\mathbf{w}_2 + \cdots + 0\mathbf{w}_k.$$

This linear combination is called the **trivial linear combination**.

If there is more than one way to express $\mathbf{0}$ as a linear combination of vectors in W, then the vectors in W are **linearly dependent**. Equivalently, if the zero vector can be written as a non-trivial linear combination of vectors in W, then the vectors in W are linearly dependent.

The phrasing "W is a set of linearly independent vectors" or "W is linearly independent" means the same as "the vectors in W are linearly independent."

Observations.

- Conventionally, the empty set is linearly independent.
- Any set W that contains $\mathbf{0}$ is linearly dependent. This follows from $\mathbf{0} = 1 \cdot \mathbf{0}$.
- If $\mathbf{w} \neq \mathbf{0}$, then the set $\{\mathbf{w}\}$ is linearly independent. To see this, suppose $\mathbf{0} = \alpha\mathbf{w}$. From theorem VI.A.2.a., $\alpha = 0$.

Theorem VI.A.3.a. The set W is linearly dependent and has at least two vectors if and only if there exists a vector $\mathbf{w} \in W$ that is a linear combination of other vectors in W.

Proof. We suppose W is linearly dependent. This means there exists a nontrivial linear combination of the zero vector:

$$\mathbf{0} = a_1\mathbf{w}_1 + a_2\mathbf{w}_2 + \cdots + a_k\mathbf{w}_k; \text{ not all } a_i \text{ are zero.}$$

Without loss of generality, suppose $a_k \neq 0$. Then

$$a_k\mathbf{w}_k = -a_1\mathbf{w}_1 - a_2\mathbf{w}_2 - \cdots - a_{k-1}\mathbf{w}_{k-1}$$

and so

$$\mathbf{w}_k = -\frac{a_1}{a_k}\mathbf{w}_1 - \frac{a_2}{a_k}\mathbf{w}_2 - \cdots - \frac{a_{k-1}}{a_k}\mathbf{w}_{k-1}$$

which establishes the forward result. The reverse direction is straightforward from definitions.

□

Corollary VI.A.3.b. Suppose $W = \{\mathbf{w}_1, \mathbf{w}_2\}$ is linearly dependent. Then there exists a scalar a such that $\mathbf{w}_1 = a\mathbf{w}_2$ or $\mathbf{w}_2 = a\mathbf{w}_1$.

Notice that the converse of corollary VI.A.3.b. is immediate, and so we have the following as a consequence, using the contrapositive.

Corollary VI.A.3.c. A set $W = \{\mathbf{w}_1, \mathbf{w}_2\}$ is linearly independent if and only if neither vector is a scalar multiple of the other.

VI.A.4. BASIS VECTORS AND DIMENSION

Definition. Given a vector space V, a set of vectors B is called a **basis** for V if

1) B is linearly independent, and
2) $V = \text{span } B$.

Example VI.A.4.a. The set

$$B = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

is a basis for $V = \mathbb{R}^3$. This is an example of a **canonical basis** (one that is easy to work with.)

Definition. For $V = \mathbb{R}^n$, the **canonical basis** consists of the set of vectors $\mathbf{e}_i$ where every entry of e_i is 0, except for the i^{th} coordinate, which is 1.

Lemma VI.A.4.b. Let V be a vector space. Suppose $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$ is a linearly independent set of vectors of V and suppose $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ is a basis for V. Then $k \leq n$.

Proof. Consider the vector $\mathbf{a}_1$ which cannot be $\mathbf{0}$ since A is linearly independent. Since B is a basis, there exists a linear combination

$$\mathbf{a}_1 = \sum_{i=1}^{n} \beta_i \mathbf{b}_i$$

and since $\mathbf{a}_1 \neq \mathbf{0}$, there must be at least one nonzero β_i. Suppose $\beta_t \neq 0$. Then

$$\mathbf{b}_t = \frac{1}{\beta_t}\left(\mathbf{a}_1 - \sum_{\substack{1 \leq i \leq n \\ i \neq t}} \beta_i \mathbf{b}_i \right)$$

and so $\mathbf{b}_t \in \text{span}(\{\mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{t-1}, \mathbf{b}_{t+1}, \dots, \mathbf{b}_n\})$. It is clear that all of the other $\mathbf{b}_i$ are in this span, so

$$\text{span}(\{\mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{t-1}, \mathbf{b}_{t+1}, \dots, \mathbf{b}_n\}) = V.$$

We now proceed inductively, supposing

$$\text{span}\left(\left\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p, \mathbf{b}_{i_1}, \dots, \mathbf{b}_{i_{n-p}}\right\}\right) = V.$$

Consider

$$\mathbf{a}_{p+1} = \sum_{i=1}^{p} \alpha_i \mathbf{a}_i + \sum_{j=1}^{n-p} \beta_{i_j} \mathbf{b}_{i_j}.$$

Since the set $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$ is linearly independent, it cannot be the case that all of the β_{i_j} are zero. Hence there is some t where $\beta_{i_t} \neq 0$ and

$$\mathbf{b}_{i_t} = \frac{1}{\beta_{i_t}} \left(\mathbf{a}_{p+1} - \sum_{i=1}^{p} \alpha_i \mathbf{a}_i - \sum_{\substack{j=1 \\ j \neq t}}^{n-p} \beta_{i_j} \mathbf{b}_{i_j} \right).$$

Thus,

$$\text{span}\left(\left\{ \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{p+1}, \mathbf{b}_{i_1}, \dots, \mathbf{b}_{i_{t-1}}, \mathbf{b}_{i_{t+1}}, \dots, \mathbf{b}_{i_{n-p}} \right\} \right) = V.$$

This process can be iterated for a total of k times, implying the set $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$ cannot have more elements than the set $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$. Hence, $k \leq n$.

▫

Theorem VI.A.4.c. If V has a finite basis, then every basis of V is the same finite cardinality.

Proof. Let $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ be a basis for V. If $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$ is another basis, then we can apply the lemma directly to obtain $k \leq n$ and switching the roles of A and B to obtain $n \leq k$, implying $k = n$. If A is an infinite basis, then it would have a finite linearly independent subset $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{n+1}\}$. Applying the lemma directly yields $n + 1 \leq n$ which is false. Hence, A cannot be infinite.

▫

Definition. The cardinality of a finite basis for a vector space V is called the **dimension** of V.

Convention. The dimension of $\{\mathbf{0}\}$ is 0; this space has the empty set as a basis.

Definition. If V has no finite basis, then V is **infinite-dimensional**.

VI.B. LINEAR MAPS AND MATRICES

VI.B.1. LINEAR MAPS

Linear maps are the fundamental functions that link vector spaces; these functions play a central role in linear algebra as a subfield of mathematics.

Definition. Given two vector spaces V, W over $\mathbb{R}$, a function $\ell: V \to W$ is a **linear map** if for all vectors $\mathbf{x}, \mathbf{y} \in V$ and scalars $\alpha \in \mathbb{R}$,

$$\ell(\mathbf{x} + \mathbf{y}) = \ell(\mathbf{x}) + \ell(\mathbf{y}),$$
$$\ell(\alpha\mathbf{x}) = \alpha\ell(\mathbf{x}).$$

This is often summarized by saying "linear maps preserve sums and scalar multiplication."

Observation. If $\ell: V \to W$ is a linear map, then for any $\mathbf{x} \in V$,

$$\ell(\mathbf{0}) = \ell(0\mathbf{x}) = 0\ell(\mathbf{x}) = \mathbf{0}.$$

This means that ℓ sends the zero vector in V to the zero vector in W. Any function that does not send the zero vector to the zero vector cannot be a linear map.

Observation. Let $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ be a basis for V and let $\ell: V \to W$ be a linear map. Then for any $\mathbf{x} \in V$, we can write

$$\mathbf{x} = \sum_{i=1}^{n} \beta_i \mathbf{b}_i$$

and, using linearity,

$$\ell(\mathbf{x}) = \ell\left(\sum_{i=1}^{n} \beta_i \mathbf{b}_i\right) = \sum_{i=1}^{n} \beta_i \, \ell(\mathbf{b}_i).$$

In brief, the behavior of a linear map is determined by its behavior on the basis vectors.

VI.B.2. RANGE AND KERNEL

Definition. Given a linear map $\ell: V \to W$, the **range** of ℓ is the set

$$\ell(V) = \{\ell\mathbf{x}: \ \mathbf{x} \in V\}$$

and the **kernel** of ℓ is the set

$$\ker(\ell) = \{\mathbf{x} \in V: \ \ell(\mathbf{x}) = \mathbf{0}\}.$$

The range is the set of all possible outputs of ℓ. This is a subspace of W.

The kernel is the set of all vectors that are sent to the zero vector in W. This is a subspace of V.

Notice that if $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is a basis for V, then

$$\ell(V) = \text{span}\{\boldsymbol{\ell}(\mathbf{b})_1, \ell(\mathbf{b}_2), \ldots, \boldsymbol{\ell}(\mathbf{b}_n)\}.$$

It is not necessarily true that $\{\boldsymbol{\ell}(\mathbf{b})_1, \ell(\mathbf{b}_2), \ldots, \boldsymbol{\ell}(\mathbf{b}_n)\}$ is a basis for W or even that this set is linearly independent.

Definition. The **rank** of a linear map $\ell: V \to W$ is the dimension of the vector space $\ell(V)$.

Definition. The **nullity** of a linear map $\ell: V \to W$ is the dimension of the kernel $\ker(\ell)$.

We state but defer the proof of the following proposition:

Proposition. The sum of the rank and nullity of a linear map $\ell: V \to W$ is the dimension of V.

VI.B.3. MATRICES

It has become standard practice to use matrices (rectangular arrays of real numbers) to represent linear maps.

Definition. Given positive integers r and c, an $r \times c$ **matrix** is a rectangular array of real numbers having r **rows** and c **columns**. If A is an $r \times c$ matrix, then the **entry** in row i and column j is denoted A_{ij}.

Definition. Given two $r \times c$ matrices A and B, their **sum** $A + B$ has entries

$$(A + B)_{ij} = A_{ij} + B_{ij}.$$

This is called the **entry-wise** sum of A and B. There is also a rarely used **entry-wise product** (also called the Hadamard product) $A * B$ whose entries are given by $(A * B)_{ij} = A_{ij}B_{ij}$.

The much more commonly used version of matrix multiplication follows:

Definition. Given an $r \times k$ matrix A and a $k \times c$ matrix B, their product AB is an $r \times c$ matrix with entries defined by the formula

$$(AB)_{ij} = \sum_{m=1}^{k} A_{im}B_{mj}.$$

It will turn out that this product arises as a result of composing linear maps. We introduce the connection between matrices and linear maps to motivate this.

Matrices are very often used to represent linear maps. Suppose $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is a basis for an n-dimensional vector space V and $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ is a basis for a k-dimensional vector space W. We consider

$$\mathbf{w}_j = \ell(\mathbf{b}_j)$$

where $\ell: V \to W$ is a linear map.

The vector $\mathbf{w}_j$ is an element of W and so there exist unique scalars $\alpha_{i,j}, 1 \leq i \leq k$ such that

$$\ell(\mathbf{b}_j) = \mathbf{w}_j = \sum_{i=1}^{k} \alpha_{i,j}\, \mathbf{c}_i.$$

Now, consider any vector $\mathbf{v} \in V$. Since $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ is a basis for V, there is a unique representation

$$\mathbf{v} = \sum_{j=1}^{n} \beta_j\, \mathbf{b}_j.$$

But then for the linear map ℓ,

$$\ell(\mathbf{v}) = \ell\left(\sum_{j=1}^{n} \beta_j\, \mathbf{b}_j\right) = \sum_{j=1}^{n} \beta_j\, \ell(\mathbf{b}_j).$$

We now use $\ell(\mathbf{b}_j) = \sum_{i=1}^{k} \alpha_{i,j}\, \mathbf{c}_i$, obtaining

$$\ell(\mathbf{v}) = \sum_{j=1}^{n} \left(\beta_j \sum_{i=1}^{k} \alpha_{i,j}\, \mathbf{c}_i\right) = \sum_{i=1}^{k} \left(\sum_{j=1}^{n} \alpha_{i,j}\beta_j\right) \mathbf{c}_i.$$

We now observe that if we define the $k \times n$ matrix A whose entries are $A_{ij} = \alpha_{i,j}$ and the $n \times 1$ matrix B whose entries are $B_{j1} = \beta_j$, then

$$(AB)_{i1} = \sum_{j=1}^{n} A_{ij} B_{j1} = \sum_{j=1}^{n} \alpha_{i,j}\beta_j.$$

VI.C. EDGE- AND VERTEX-INDEXED MATRICES

We now introduce various matrices that are commonly used in connection with graphs. These include the incidence matrix, adjacency matrix, the degree diagonal matrix, and the Laplacian matrix.

VI.C.1. INCIDENCE MATRICES

Definition. Given a graph G with vertex set $V = \{v_1, v_2, \dots, v_n\}$ and edge set $E = \{e_1, e_2, \dots, e_m\}$, we form its **incidence matrix** B, an $n \times m$ matrix whose entries are

$$B_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is an endpoint of } e_j; \\ 0, & \text{otherwise.} \end{cases}$$

Notice that the rows of the matrix correspond to vertices and the columns correspond to edges.

Common Conventions.

- If e_j is a loop with both endpoints v_i, then $B_{ij} = 1$. This is often adopted so that constructions of the adjacency and Laplacian matrices remain somewhat consistent.
- If e_j is a loop with both endpoints v_i, then $B_{ij} = 2$. This is often adopted so that the columns all sum to 2.

Our examples will not consider graphs with loops, avoiding this issue entirely. Parallel edges will be allowed.

For instance, let G be the graph depicted in figure VI.C.1.a.

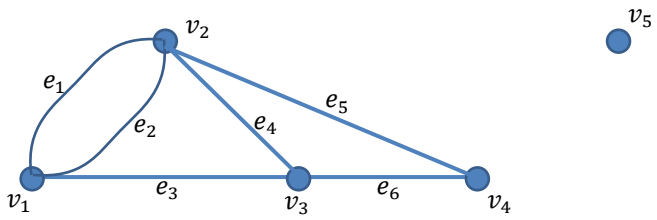


Figure VI.C.1.a. An example graph G.

The incidence matrix for this graph (as indexed) is

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Observations (when the graph G has no loops).

- Two graphs are isomorphic if and only if the incidence matrix of one can be obtained by permuting the rows and columns of the other.
- Every column sums to 2.
- Every row sums to the degree of the corresponding vertex.

- The dot-product of two different rows equals the number of edges incident with the two vertices involved.
- The dot-product of two different columns is 0 if the edges share no endpoints, 1 if the edges share one endpoint, and 2 if the edges are parallel.
- An all-zero row corresponds to an isolated vertex.

For loopless digraphs, we amend the definition of B slightly. We agree to use B' rather than B when discussing digraphs.

$$B'_{ij} = \begin{cases} 1, & \text{if arc } e_j \text{ leaves node } v_i; \\ -1, & \text{if arc } e_j \text{ enters node } v_i; \\ 0, & \text{otherwise.} \end{cases}$$

Consider the digraph G' in figure VI.C.1.b.

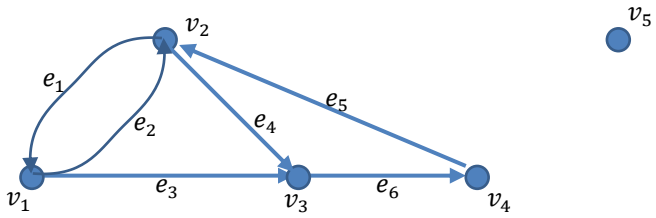


Figure VI.C.1.b. An example digraph G'

This digraph's incidence matrix is

$$B' = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Observations when the digraph *D* has no loops.

- Two digraphs are isomorphic if and only if the incidence matrix of one can be obtained by permuting the rows and columns of the other.
- Every column sums to 0.
- Every row sums to $d_{\text{out}}(v_i) - d_{\text{in}}(v_i)$ where v_i is the node corresponding to that row.
- The dot-product of a row with itself is the total degree of the corresponding node.
- The dot-product of two distinct rows is negative the number of arcs joining the corresponding nodes.
- An all-zero row corresponds to an isolated node.

VI.C.2. ADJACENCY, DEGREE-DIAGONAL, AND LAPLACIAN MATRICES DEFINED

The incidence matrix can be used to define several other matrices commonly associated with graphs:

Definition. Let G be a loopless graph and B be its incidence matrix. Recall that B^T is its transpose. Consider the matrix

$$M = BB^T.$$

The **degree-diagonal matrix** D is the diagonal matrix whose entries are the diagonal entries of M.

The **adjacency matrix** A is the matrix $M - D$.

We can write $M = A + D$.

All of this captures the various observations of the incidence matrix concerning dot-products.

For instance, if

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

then

$$M = BB^T = \begin{bmatrix} 3 & 2 & 1 & 0 & 0 \\ 2 & 4 & 1 & 1 & 0 \\ 1 & 1 & 3 & 1 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$D = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad A = \begin{bmatrix} 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

If G is simple, then the entries of A are either 0 or 1.

If one orients G in any fashion, turning it into a digraph G', with resulting incidence matrix B', then

$$M' = B'(B')^T = D - A.$$

Definition. The **Laplacian matrix** of G is the matrix $L = D - A$.

In our example,

$$B' = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$L = M' = B'(B')^T = \begin{bmatrix} 3 & -2 & -1 & 0 & 0 \\ -2 & 4 & -1 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Observation. All three of the matrices A, D, L are symmetric matrices with real entries.

VI.C.3. PERMUTATION MATRICES

Definition. A **permutation** of a set S is a bijection $\sigma: S \to S$.

Definition. An n-**permutation** is a permutation over $\{1,2,\dots,n\}$.

Definition. A **permutation matrix** over $\{1,2,\dots,n\}$ is an $n \times n$ matrix P where all entries are zero except that in each row and in each column, there is a single 1 entry.

Definition. If σ is an n-permutation, then P_σ is the $n \times n$ permutation matrix such that

$$P_\sigma \, \mathbf{e}_i = \mathbf{e}_{\sigma(i)}$$

where $\mathbf{e}_i$ is the i^{th} standard basis vector, i.e., the vector with all zero entries, except for a 1 in position i.

It follows that

$$(P_\sigma)_{ij} = \begin{cases} 1, & j = \sigma(i); \\ 0, & j \neq \sigma(i). \end{cases}$$

Examples. The matrices below are permutation matrices:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Observations.

- Identity matrices are permutation matrices.
- Permutation matrices are orthogonal, i.e., if P is a permutation matrix, then $PP^T = I$, or $P^T = P^{-1}$.
- If P is a permutation matrix and PA is defined, then the rows of PA are a permutation of the rows of A.
- If P is a permutation matrix and AP is defined, then the columns of AP are a permutation of the columns of A.

Definition. Let G be a graph with vertices $V = \{v_1, v_2, \dots, v_n\}$ and edges $E = \{e_1, e_2, \dots, e_m\}$. If σ is an n-permutation, then **permuting the vertices of G by σ** results in the graph G_σ with the same vertex set $V' = V$, but where there is a bijection between E and E' where if $e_i = v_s v_t$, then $e_i' = v_{\sigma(s)} v_{\sigma(t)}$. Essentially, the effect is to replace each label v_i with $v_{\sigma(i)}$.

Observation. If B is the incidence matrix for the graph G, then $P_\sigma B$ is the incidence matrix for the graph G_σ. This can be seen by considering each column of B separately, noticing that

$$\mathbf{b}_i = \mathbf{e}_s + \mathbf{e}_t$$

where $\mathbf{b}_i$ is the i^{th} column of B. Hence,

$$(P_\sigma B)_i = P_\sigma \mathbf{b}_i = P_\sigma(\mathbf{e}_s + \mathbf{e}_t) = \mathbf{e}_{\sigma(s)} + \mathbf{e}_{\sigma(t)},$$

the column representing an edge joining $v_{\sigma(s)}$ and $v_{\sigma(t)}$.

Observation. Since $P_\sigma B$ is the incidence matrix for G_σ, $M' = (P_\sigma B)(P_\sigma B)^T = A' + D'$, where A' is the adjacency matrix for G_σ and D' is the degree-diagonal matrix for G_σ. Hence, if $M = BB^T = A + D$, we have

$$A' + D' = (P_\sigma B)(P_\sigma B)^T = P_\sigma B B^T P_\sigma^T = P_\sigma M P_\sigma^T = P_\sigma (A + D) P_\sigma^T = P_\sigma A P_\sigma^T + P_\sigma D P_\sigma^T.$$

Observation. For any $i \in \{1,2,\dots,n\}$,

$$\begin{aligned} P_\sigma D P_\sigma^T \, \mathbf{e}_{\sigma(i)} &= P_\sigma D \, \mathbf{e}_i \\ &= P_\sigma(\deg_G(v_i)\, \mathbf{e}_i) \\ &= \deg_G(v_i)\; P_\sigma \mathbf{e}_i \\ &= \deg_{G_\sigma}\left(v_{\sigma(i)}\right)\; \mathbf{e}_{\sigma(i)} \\ &= D'_{\sigma(i),\sigma(i)} \end{aligned}$$

and so $D' = P_\sigma D P_\sigma^T$. We then have $A' = P_\sigma A P_\sigma^T$ as a consequence.

Definition. If there exists a permutation matrix P such that $M' = PMP^T$, then M and M' are **permutation similar** matrices.

We have the following result as a consequence of the preceding discussion:

Proposition. Two finite graphs are isomorphic if and only if their adjacency matrices are permutation similar.

VI.C.4. MATRICES OF BIPARTITE GRAPHS

Suppose G is a bipartite graph with vertex bipartition $V = V_1 \sqcup V_2$. If we index the vertices as follows

$$V_1 = \{v_1, v_2, \dots, v_k\};\;\; V_2 = \{v_{k+1}, v_{k+2}, \dots, v_n\}$$

then the adjacency matrix for G will take the form

$$A = \begin{bmatrix} O & B \\ B^T & O' \end{bmatrix}$$

where B is a $k \times (n-k)$ matrix over $\{0,1\}$, O is the $k \times k$ all-zero matrix, and O' is the $(n-k) \times (n-k)$ all-zero matrix.