Lecture Notes: Graph Theory, Part 1

Graphs, trails and paths, connectivity, subgraphs, complements, isomorphism and vertex degree

COMP 339 lecture notes, textbook Sections 11.1 to 11.3 (by Parsa Kamalipour)

What you will be able to do after this lecture

1. State what a **directed graph** and an **undirected graph** are, and use the words **adjacent**, **incident** and **loop** correctly.
2. Decide whether a sequence of vertices and edges is a **trail**, a **circuit**, a **path** or a **cycle**.
3. Decide whether a graph is **connected**, and prove that a trail between two vertices can always be shortened to a path.
4. Recognise **subgraphs**, **spanning subgraphs**, the **complete graph** K_n and the **complement** $\overline{G}$ of a graph.
5. Prove that two graphs are **isomorphic** by writing down a correspondence and checking every edge, and prove that two graphs are **not** isomorphic by finding a property that one has and the other does not.
6. Compute the **degree** of a vertex, and the **in-degree** and **out-degree** in a directed graph, counting loops correctly.

Each topic comes with the intuition, a simple example, the formal definition, a comprehensive example and a recap. Section 4 collects the writing templates, the classic mistakes and exercises. **Reading:** textbook Sections 11.1 to 11.4.

Contents

1 Graphs, trails and paths (Section 11.1) 2
1.1 Directed and undirected graphs . . . 2
1.2 Trails, circuits, paths and cycles . . . 3
1.3 Connected graphs . . . 5

2 Subgraphs, complements and graph isomorphism (Section 11.2) 6
2.1 Subgraphs and spanning subgraphs . . . 6
2.2 Complete graphs and complements . . . 8
2.3 Graph isomorphism . . . 10
2.4 Showing that two graphs are not isomorphic . . . 13

3 Vertex degree (beginning of Section 11.3) 14

4 Templates, mistakes and exercises 16
4.1 Writing templates . . . 16
4.2 Common mistakes . . . 17
4.3 Exercises . . . 17

A Short answers to the exercises 17

1 Graphs, trails and paths (Section 11.1)

1.1 Directed and undirected graphs

The intuition

A graph is a set of **dots** (vertices) together with **lines** between some pairs of dots (edges). That is all: the position of the dots, the length of the lines and whether lines cross in the drawing carry **no information**. The only thing a graph records is *which pairs of vertices are joined*.
Think of a road map. If every street is **one-way**, an edge goes *from* one vertex *to* another, and we record it as an ordered pair (u, v): this is a **directed graph**. If every street is **two-way**, the direction does not matter and we record the edge as the set $\{u, v\}$: this is an **undirected graph**.



Figure 1. Left: the directed graph with $E = \{(a,b),(b,a),(b,d),(c,d),(c,a),(d,d)\}$. Right: an undirected graph on the same vertices with $E = \{\{a,b\},\{b,d\},\{c,d\},\{a,c\}\}$.

Simple example: reading a graph from its sets

Let $V = \{a,b,c,d\}$ and $E = \{\{a,b\},\{b,d\},\{c,d\},\{a,c\}\}$. Draw four dots and one line for each set in E: you get the square on the right of Figure 1. The pair $\{a,d\}$ is not in E, so a and d are not joined, even if a different drawing puts them close together.

Directed graph, undirected graph

Let V be a finite set, and let $E \subseteq V \times V$. The pair (V,E) is called a **directed graph** or **digraph**, where

- V is the set of **vertices** or **nodes**, and
- E is the set of (directed) **edges** or **arcs**.

We denote such a graph by $G = (V,E)$. When the direction of the edges does not matter, G is called an **undirected graph**, and an edge between a and b is written $\{a,b\}$.
Convention. From now on, when we say "a graph" we mean an **undirected** graph.

Adjacent, incident, loop

If $u \in V$, $v \in V$ and $e = (u,v) \in E$, then

- the vertices u and v are **adjacent**;
- the edge $e = (u,v)$ is **incident** with u and v;
- an edge of the form $e = (v,v)$ is called a **loop**.

A graph with no loops is called **loop-free**.

Comprehensive example: the directed graph of Figure 1

- (a, b) and (b, a) are **two different arcs**: one goes from a to b, the other from b to a.
- b and d are adjacent because $(b, d) \in E$. The arc (b, d) is incident with b and with d.
- a and d are **not** adjacent: neither (a, d) nor (d, a) is in E.
- (d, d) is a loop; it is incident with d only. So this digraph is not loop-free.
- $|V| = 4$ and $|E| = 6$. Since $E \subseteq V \times V$ and $|V \times V| = 16$, a digraph on 4 vertices has at most 16 arcs (loops included).

Ordered pair or set?

In a directed graph $(u, v) \neq (v, u)$ when $u \neq v$: the order is the direction. In an undirected graph $\{u, v\} = \{v, u\}$: there is only one edge between u and v. Writing (a, b) for an undirected edge is a common slip; use braces.

Graphs in one breath

- Digraph: $G = (V, E)$ with $E \subseteq V \times V$; arcs are ordered pairs. Graph: edges are sets $\{a, b\}$.
- Adjacent: joined by an edge. Incident: the edge touches the vertex. Loop: an edge from a vertex to itself.

1.2 Trails, circuits, paths and cycles

The intuition

Imagine walking through the graph, going from vertex to vertex along edges, and writing down everything you visit: vertex, edge, vertex, edge, ..., vertex. Two rules give the four words of this section:

- a **trail** never uses the same **edge** (street) twice;
- a **path** never visits the same **vertex** (intersection) twice.

If the walk ends where it started, a trail becomes a **circuit** and a path becomes a **cycle**.

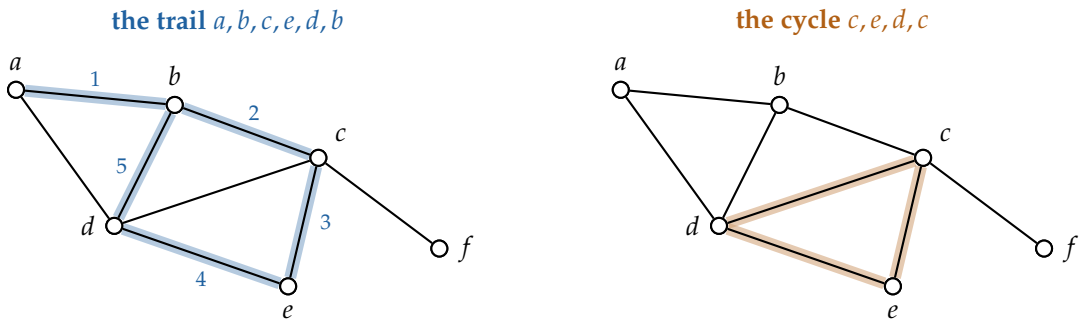


Figure 2. The graph from lecture, with edges $\{a, b\}, \{b, c\}, \{c, f\}, \{a, d\}, \{b, d\}, \{c, d\}, \{c, e\}, \{d, e\}$. Left: a trail that is not a path (it visits b twice). Right: a circuit that is also a cycle.

Simple example: a short path

In Figure 2, the sequence $a, \{a,b\}, b, \{b,c\}, c, \{c,f\}, f$ uses three different edges and four different vertices a, b, c, f. No edge and no vertex repeats, so it is an a-f **path** of length 3.

Trail, circuit, path, cycle

Let u and v be (not necessarily distinct) vertices in an undirected graph $G = (V, E)$. A finite sequence

$$u = u_0,\ e_1,\ u_1,\ e_2,\ u_2,\ \ldots,\ e_{n-1},\ u_{n-1},\ e_n,\ u_n = v$$

of vertices and edges from G, where $e_i = \{u_{i-1}, u_i\}$ for $1 \le i \le n$, and **no edge is repeated** ($e_i \neq e_j$ for $i \neq j$), is called a u-v **trail**. The number n of edges is its **length**.

- A u-u trail (it ends where it starts) is called a **circuit**.
- If no vertex occurs more than once in the sequence, the trail is called a u-v **path**.
- When $u = v$ (and no other vertex repeats), it is called a **cycle**.

Similarly we define **directed** trails, circuits, paths and cycles in a directed graph, where now $e_i = (u_{i-1}, u_i)$: every arc is followed in its own direction.

Name	Ends where it starts?	Edge may repeat?	Vertex may repeat?
trail	either	no	yes
circuit	yes ($u = v$)	no	yes
path	no ($u \neq v$)	no	no
cycle	yes ($u = v$)	no	only $u_0 = u_n$

Table 1. The four kinds of sequence. Every path is a trail and every cycle is a circuit, but not conversely.

A word you will see in the textbook: walk

If we drop the condition "no edge is repeated", the sequence is called a u-v **walk**. So walk $\supseteq$ trail $\supseteq$ path. In a graph without parallel edges the edges are determined by the vertices, so we often write just the vertices: a, b, c, e, d, b.

Comprehensive example: classifying sequences in Figure 2

(1) $a\ \{a,b\}\ b\ \{b,c\}\ c\ \{c,e\}\ e\ \{e,d\}\ d\ \{d,b\}\ b$.
The five edges are all different, so this is an a-b **trail** of length 5. It is **not a path**: the vertex b occurs twice.

(2) $c\ \{c,e\}\ e\ \{e,d\}\ d\ \{d,c\}\ c$.
Three different edges, and it returns to c: a **circuit**. Apart from c at both ends no vertex repeats, so it is **also a cycle** (of length 3).

(3) b, a, d, c, e, d, b, that is $b\ \{b,a\}\ a\ \{a,d\}\ d\ \{d,c\}\ c\ \{c,e\}\ e\ \{e,d\}\ d\ \{d,b\}\ b$.
Six different edges, back to b: a **circuit**. But d occurs twice in the middle, so it is **not a cycle**.

(4) a, b, d, a, b.
The edge $\{a,b\}$ is used twice: **not even a trail** (only a walk).

(5) No cycle passes through f: f has only one edge, $\{c,f\}$, and a cycle through f would have to leave f along a second edge.

Two slips with these words

- **Checking the wrong thing.** "No edge repeats" (trail) and "no vertex repeats" (path) are different tests. Sequence (1) above passes the first and fails the second.
- **Using an edge that is not there.** Every consecutive pair u_{i-1}, u_i must be an edge of G. In Figure 2, a, e is not allowed because $\{a, e\} \notin E$.

Trails and paths in one breath

- Trail: no repeated edge. Path: no repeated vertex. Closed trail: circuit. Closed path: cycle.
- Every path is a trail; every cycle is a circuit. Length = number of edges.

1.3 Connected graphs

The intuition

A graph is connected when it is "in one piece": from any vertex you can walk to any other vertex. If the graph falls apart into two or more pieces with no edges between them, it is disconnected.

Connected, disconnected

A graph $G = (V, E)$ is called **connected** if there is a path between any two distinct vertices of G. A graph that is not connected is called **disconnected**.

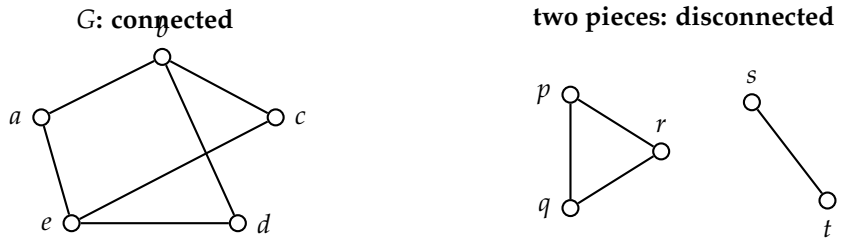


Figure 3. Left: a connected graph (the graph G from lecture). Right: a disconnected graph; there is no path between p and s.

Simple example: checking connectivity

In the left graph of Figure 3, there are paths from a to every other vertex: a, b; a, b, c; a, b, d; a, e. But the definition asks for a path between *every* pair, for example c and d. Going through a gives c, b, a, b, d, which repeats b; here c, e, d works directly. Below we show that reaching every vertex from one vertex is always enough.

The following theorem was left as an exercise in lecture; here is the proof.

Theorem (trails can be shortened to paths)

Let $G = (V, E)$ be an undirected graph and let $a, b \in V$ with $a \neq b$. If there exists a trail (in G) from a to b, then there exists a path (in G) from a to b.

Proof

Solution (what you write) | **Commentary (what it does)**

Statement. If there is an a-b trail in G, $a \neq b$, then there is an a-b path in G.

Proof. Among all a-b trails in G choose one of **smallest length** n:

$$a = u_0,\ e_1,\ u_1,\ \ldots,\ e_n,\ u_n = b.$$

There is at least one trail, so a shortest one exists (lengths are non-negative integers).

Suppose, for a contradiction, that some vertex occurs twice: $u_i = u_j$ with $i < j$. Delete the part between the two occurrences:

$$a = u_0,\ e_1,\ \ldots,\ e_i,\ u_i,\ e_{j+1},\ u_{j+1},\ \ldots,\ e_n,\ u_n = b.$$

Cut out the "loop" $u_i, e_{i+1}, \ldots, e_j, u_j$ that comes back to the same vertex.

This is still an a-b trail: $e_{j+1} = \{u_j, u_{j+1}\} = \{u_i, u_{j+1}\}$, so consecutive terms still match, and its edges are some of the original edges, so none repeats. Its length is $n - (j - i) < n$. This contradicts the choice of a shortest trail.

Removing edges can never create a repeated edge.

Hence no vertex occurs twice, and the shortest trail is an a-b **path**. ■

Comprehensive example: the shortcut in action, and connectivity made easy

(i) In Figure 2, a, b, c, e, d, b is an a-b trail in which b occurs at positions 1 and 5. Cutting out b, c, e, d, b leaves $a, \{a, b\}, b$: an a-b path of length 1.
(ii) *To show a graph is connected it is enough to reach every vertex from one fixed vertex x.* Indeed, for distinct u, v, follow a u-x path backwards and then an x-v path. This is a u-v walk (it may repeat edges). The proof of the theorem works word for word for walks (a shortest u-v walk cannot repeat a vertex), so it can be shortened to a u-v path. For the left graph of Figure 3 the four paths from a in the simple example are therefore enough: G **is connected**.

Connectivity in one breath

- Connected: a path between any two distinct vertices. Otherwise disconnected.
- A trail from a to b can always be shortened to a path (cut out the part between two visits of the same vertex).

2 Subgraphs, complements and graph isomorphism (Section 11.2)

2.1 Subgraphs and spanning subgraphs

The intuition

A subgraph is what you get by **erasing** parts of a drawing: erase some edges, and possibly some vertices (but if you erase a vertex you must also erase the edges touching it, since an edge cannot hang in the air). You may never **add** anything. If you keep **all** the vertices and only erase edges, the subgraph is **spanning**.

Subgraph, spanning subgraph

If $G = (V, E)$ is a graph (or digraph), then $G_1 = (V_1, E_1)$ is called a **subgraph** of G if

$$\emptyset \neq V_1 \subseteq V \quad \text{and} \quad E_1 \subseteq E,$$

where each edge in E_1 is incident with vertices in V_1 (both of its ends lie in V_1).
Given a graph (digraph) $G = (V, E)$, let $G_1 = (V_1, E_1)$ be a subgraph of G. If $V_1 = V$, then G_1 is called a **spanning subgraph** of G.

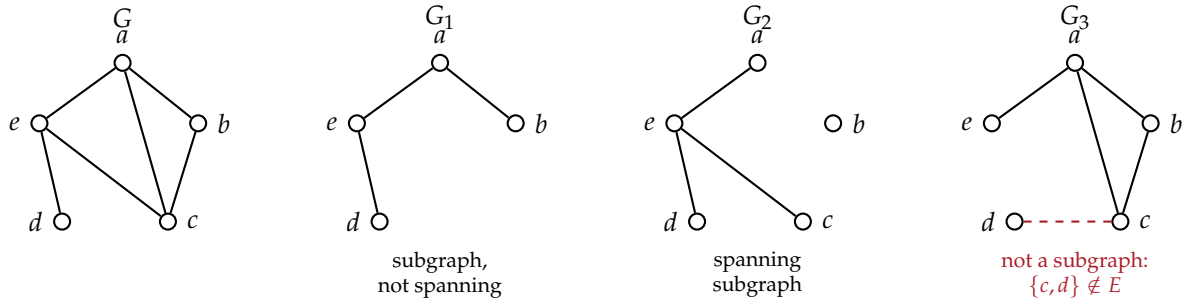


Figure 4. The example from lecture. G_1 and G_2 only erase things from G. G_3 contains the edge $\{c, d\}$ (dashed), which is not an edge of G.

Simple example: erasing edges

Every graph is a spanning subgraph of itself. Erasing any set of edges (and no vertices) from G always gives a spanning subgraph. In Figure 4, G_2 is G with $\{a, b\}, \{a, c\}, \{b, c\}$ erased: b is still there, now with no edges.

Comprehensive example: Figure 4 and a directed version

- $G_1 = (\{a, b, d, e\}, \{\{e, a\}, \{a, b\}, \{e, d\}\})$. Here $V_1 \subseteq V$, $E_1 \subseteq E$, and every edge of E_1 has both ends in V_1. So G_1 **is a subgraph** of G, **but not a spanning** subgraph since $c \notin V_1$.
- G_2 has all five vertices and $E_2 = \{\{e, a\}, \{e, c\}, \{e, d\}\} \subseteq E$: a **spanning subgraph**.
- $G_3 = (V_3, E_3)$ is **not a subgraph** of G, because $\{c, d\} \in E_3$ but $\{c, d\} \notin E$, so $E_3 \not\subseteq E$.

Directed graphs. Let $H = (V, E)$ be the digraph below. Then $H_1 = (V_1, E_1)$ with $V_1 = \{v, u, y\}$ and $E_1 = \{(v, y), (v, u), (u, y)\}$ is a subgraph of H: each of these arcs is an arc of H, with the same direction.

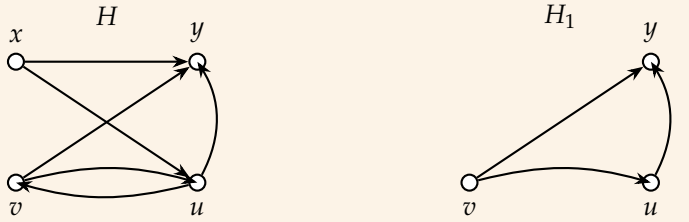


Figure 5. The digraph H with arcs $(x, y), (x, u), (v, y), (v, u), (u, v), (u, y)$ and its subgraph H_1.

Two slips with subgraphs

- **An edge whose end was erased.** $V_1 = \{a, b\}$, $E_1 = \{\{a, e\}\}$ is *not* a subgraph of G: $\{a, e\}$ is incident with $e \notin V_1$.
- **Same picture, different edges.** A subgraph must use edges *of* G. Before checking anything else, check $E_1 \subseteq E$ edge by edge, as for G_3. In a digraph, $(u, v) \in E$ does not give you (v, u).

Subgraphs in one breath

- Subgraph: $\emptyset \neq V_1 \subseteq V$, $E_1 \subseteq E$, every edge of E_1 has its ends in V_1.
- Spanning: in addition $V_1 = V$ (only edges were erased).

2.2 Complete graphs and complements

The intuition

The **complete graph** K_n has every possible edge: it is "everyone knows everyone". Any loop-free graph on n vertices is obtained from K_n by erasing some edges. The **complement** $\overline{G}$ keeps the same vertices and **swaps** edges and non-edges: an edge of K_n is in $\overline{G}$ exactly when it is not in G.

Complete graph

Let V be a set of n vertices. The **complete graph** on V, denoted K_n, is a loop-free undirected graph where for all $a, b \in V$, $a \neq b$, there is an edge $\{a, b\}$.

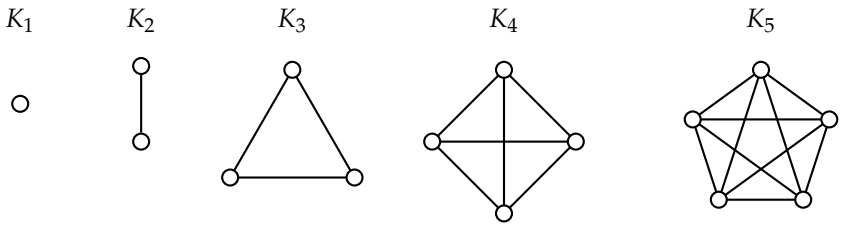


Figure 6. The complete graphs K_1 to K_5. K_n has one edge for every 2-subset of the vertices, so $\binom{n}{2}$ edges: $0, 1, 3, 6, 10$.

Complement

Let G be a loop-free (undirected) graph on n vertices. The **complement** of G, denoted $\overline{G}$, is the subgraph of K_n consisting of the n vertices in G and all edges that are not in G.
Consequence. Every edge of K_n is in exactly one of G and $\overline{G}$, so $|E(G)| + |E(\overline{G})| = \binom{n}{2}$.

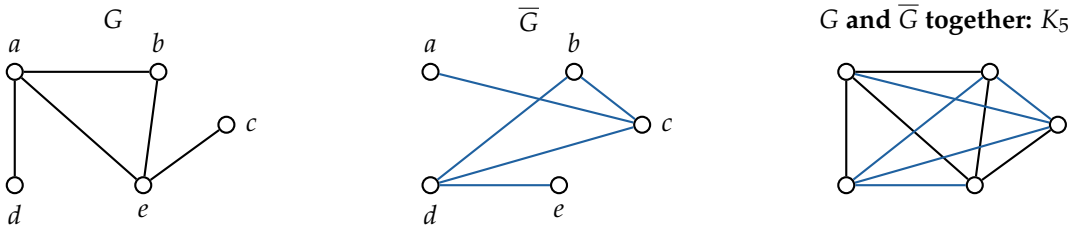


Figure 7. The example from lecture, drawn with the vertices in the same places. (The lecture drawing of $\overline{G}$ moves the vertices around; it is the same graph.) Black and blue edges together give K_5.

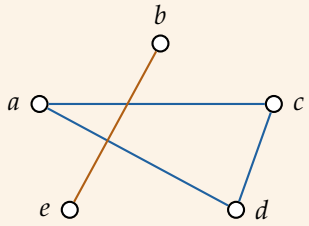
Simple example: the complement of K_4

$G = K_4$ already has all $\binom{4}{2} = 6$ edges, so $\overline{K_4}$ has 4 vertices and **no edges**. A graph with vertices but no edges is called a **null graph**. In general, the complement of K_n is the null graph on n vertices.

Comprehensive example: G connected, $\overline{G}$ not connected

Take the connected graph G of Figure 3 (left), with edges $\{a,b\}, \{b,c\}, \{b,d\}, \{a,e\}, \{e,d\}, \{e,c\}$. The 10 pairs of K_5 minus these 6 edges leave 4 edges for the complement:

$$E(\overline{G}) = \{\{a,c\}, \{a,d\}, \{c,d\}, \{b,e\}\}.$$



$\overline{G}$ is **not connected**: it splits into the pieces $\{a,c,d\}$ and $\{b,e\}$, and there is no path between, say, b and c. *Check:* $6 + 4 = 10 = \binom{5}{2}$. ✓

Remember. If G is a connected graph, then $\overline{G}$ (the complement of G) is **not necessarily** connected.

Complement is taken inside K_n on the same vertices

The complement keeps **all n vertices**, even those that end up with no edges, and it never contains loops. Erasing the isolated vertices of $\overline{G}$ (as in $\overline{K_4}$) gives a different graph.

K_n and complements in one breath

- K_n: all $\binom{n}{2}$ edges. $\overline{G}$: same vertices, exactly the edges of K_n that G lacks.
- $|E(G)| + |E(\overline{G})| = \binom{n}{2}$; $\overline{K_n}$ is the null graph; G connected does not imply $\overline{G}$ connected.

2.3 Graph isomorphism

The intuition

The same graph can be drawn in many ways. Two drawings show **isomorphic** graphs when we can **rename** the vertices of one so that it becomes exactly the other: same vertices (after renaming), and the same pairs joined. The renaming is a function f that preserves adjacency. The only essential property is **which pairs of vertices are adjacent and which are not**; the structure of the graph is preserved.

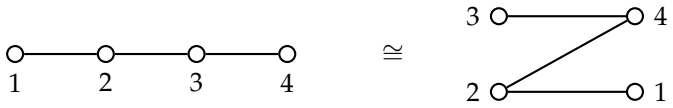


Figure 8. A path on four vertices drawn straight and drawn as a zigzag (Example 1 from lecture). Both have the edges $\{1,2\},\{2,3\},\{3,4\}$; only the drawing differs.

Graph isomorphism

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two undirected graphs. A function $f : V_1 \to V_2$ is called a **graph isomorphism** if

(a) f is **one-to-one and onto** (onto: for every $u \in V_2$ there is at least one $v \in V_1$ with $f(v) = u$), and

(b) for all $a, b \in V_1$: $\{a,b\} \in E_1$ **if and only if** $\{f(a), f(b)\} \in E_2$.

When such a function exists, G_1 and G_2 are called **isomorphic graphs**. The vertex correspondence of a graph isomorphism **preserves adjacencies**.

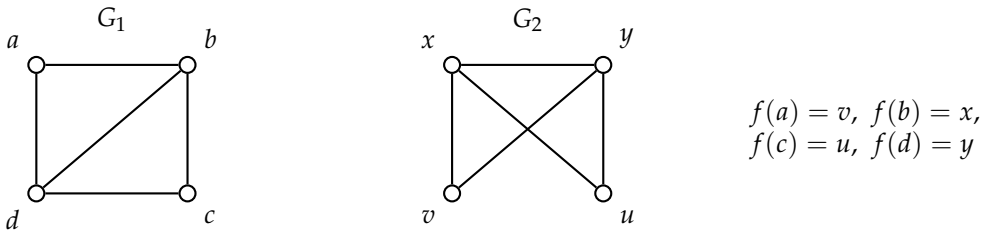


Figure 9. Example 2 from lecture. In G_1 the diagonal is $\{b,d\}$; in G_2 the two diagonals cross, but only $\{x,u\}$ and $\{v,y\}$ are edges.

Example 2: f is a graph isomorphism. Why?

Solution (what you write)	**Commentary (what it does)**
Statement. Let $f : V_1 \to V_2$ be defined by $f(a) = v$, $f(b) = x$, $f(c) = u$, $f(d) = y$. Then f is a graph isomorphism from G_1 to G_2.	*To prove "isomorphic", give f explicitly.*
(a) The images v, x, u, y are four different vertices and they are all of V_2, so f is one-to-one and onto.	

(b) Every edge of G_1 goes to an edge of G_2:

$$\{a,b\} \leftrightarrow \{f(a),f(b)\} = \{v,x\}$$
$$\{b,c\} \leftrightarrow \{f(b),f(c)\} = \{x,u\}$$
$$\{c,d\} \leftrightarrow \{f(c),f(d)\} = \{u,y\}$$
$$\{d,a\} \leftrightarrow \{f(d),f(a)\} = \{y,v\}$$
$$\{b,d\} \leftrightarrow \{f(b),f(d)\} = \{x,y\}$$

These are 5 different edges of G_2, and G_2 has exactly 5 edges, so every edge of G_2 is hit. Hence $\{a,b\} \in E_1 \iff \{f(a),f(b)\} \in E_2$ for all a,b. ■

The "if" direction: since f is one-to-one, different edges go to different edges; with $|E_1| = |E_2|$ the list above is all of E_2, so non-edges go to non-edges.

G_1 is connected. What about G_2? G_2 is also connected. This is not a coincidence:

Theorem (isomorphism preserves connectivity)

If G_1 and G_2 are isomorphic graphs, then: if G_1 is connected, G_2 is connected; and if G_1 is disconnected, G_2 is disconnected.

Proof

Solution (what you write)	**Commentary (what it does)**
Let $f: V_1 \to V_2$ be an isomorphism and suppose G_1 is connected. Take distinct $p,q \in V_2$. Since f is onto and one-to-one, $p = f(a)$ and $q = f(b)$ for distinct $a,b \in V_1$.	*Pull the two vertices back to G_1.*
G_1 is connected, so it has a path $a = u_0,u_1,\ldots,u_n = b$. Then $f(u_0),f(u_1),\ldots,f(u_n)$ is a path from p to q in G_2: each $\{f(u_{i-1}),f(u_i)\} \in E_2$ by (b), and the vertices are distinct because f is one-to-one.	*Push the path forward with f.*
So G_2 is connected. Applying the same argument to f^{-1} (also an isomorphism) shows: G_2 connected $\Rightarrow$ G_1 connected. Taking the contrapositive gives the statement for "disconnected". ■	

How do you come up with the correspondence?

Isomorphism preserves graph **substructures** such as paths and cycles (the proof above shows it for paths). So look for a structure in G_1, find the same structure in G_2, and match them. The comprehensive example below was the homework from lecture.

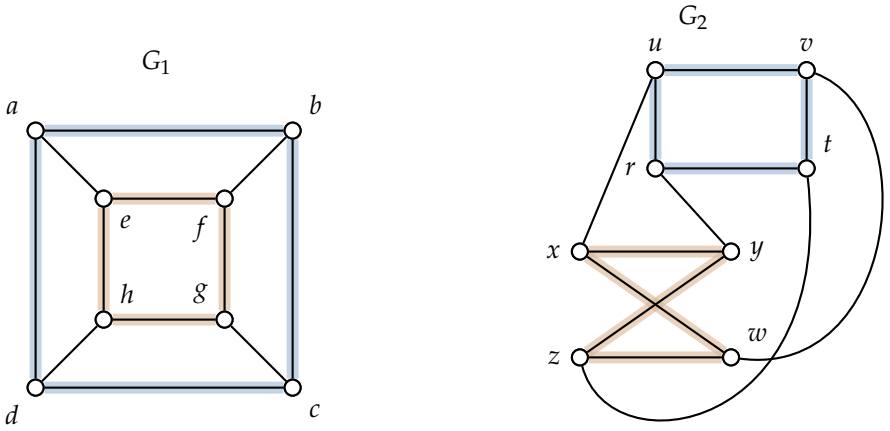


Figure 10. The homework example from lecture. Blue: a cycle of length 4 in each graph. Orange: a second cycle of length 4, disjoint from the first. The remaining four edges join the two cycles one-to-one.

Comprehensive example: G_1 and G_2 of Figure 10 are isomorphic

Solution (what you write) | **Commentary (what it does)**

Statement. G_1 and G_2 in Figure 10 are isomorphic.

Both have 8 vertices and 12 edges, and every vertex has 3 neighbours: no obvious obstruction.

Finding f. In G_1, a, b, c, d (edges $\{a,b\},\{b,c\},\{c,d\},\{d,a\}$) is a cycle of length 4. So we look for a cycle of length 4 in G_2: u, v, t, r (edges $\{u,v\},\{v,t\},\{t,r\},\{r,u\}$). Similarly e, f, g, h in G_1 matches x, w, z, y in G_2.

Isomorphism preserves cycles, so cycles must go to cycles of the same length.

In both graphs we have 2 cycles of length 4 and a **perfect matching** (a one-to-one pairing) between them:

$$\{a,e\} \to \{u,x\}, \quad \{b,f\} \to \{v,w\},$$
$$\{c,g\} \to \{t,z\}, \quad \{d,h\} \to \{r,y\}.$$

The matching fixes how the second cycle must be lined up with the first.

This suggests

$$\begin{array}{llll} a \to u & c \to t & e \to x & g \to z \\ b \to v & d \to r & f \to w & h \to y \end{array}$$

Read the two blue cycles in the same order, and the two orange cycles in the order forced by the matching.

Checking (a). The 8 images are the 8 different vertices of G_2: f is one-to-one and onto.

Checking (b). The 12 edges of G_1 go to

$$\begin{array}{lll} \{a,b\} \to \{u,v\} & \{e,f\} \to \{x,w\} & \{a,e\} \to \{u,x\} \\ \{b,c\} \to \{v,t\} & \{f,g\} \to \{w,z\} & \{b,f\} \to \{v,w\} \\ \{c,d\} \to \{t,r\} & \{g,h\} \to \{z,y\} & \{c,g\} \to \{t,z\} \\ \{d,a\} \to \{r,u\} & \{h,e\} \to \{y,x\} & \{d,h\} \to \{r,y\} \end{array}$$

Column 1: the blue cycle. Column 2: the orange cycle. Column 3: the matching.

all of which are edges of G_2. They are 12 different edges and G_2 has 12 edges, so non-edges go to non-edges. Hence f is a graph isomorphism and $\boxed{G_1 \cong G_2}$. ■

"They look different" is not a proof, and neither is "they look alike"

Drawings can be misleading in both directions (Figures 8 and 10). To prove two graphs **are** isomorphic you must give f and check **every** edge. Trying one function f that fails proves nothing: another f might work. To prove they are **not** isomorphic, see the next subsection.

Isomorphism in one breath

- $f\colon V_1 \to V_2$ one-to-one and onto, with $\{a,b\} \in E_1 \iff \{f(a),f(b)\} \in E_2$.
- To find f: match structures (cycles, paths, special vertices). To verify f: list every edge and its image.
- Isomorphism preserves connectivity, paths and cycles.

2.4 Showing that two graphs are not isomorphic

The intuition

If G_1 and G_2 are isomorphic, every property that is defined only in terms of "who is adjacent to whom" is the same for both. Such a property is called an **invariant**. So to prove two graphs are **not** isomorphic, find an invariant on which they **differ**. Number of vertices, number of edges and connectivity are invariants. Another one, used below, is **how many other vertices a vertex is adjacent to**.

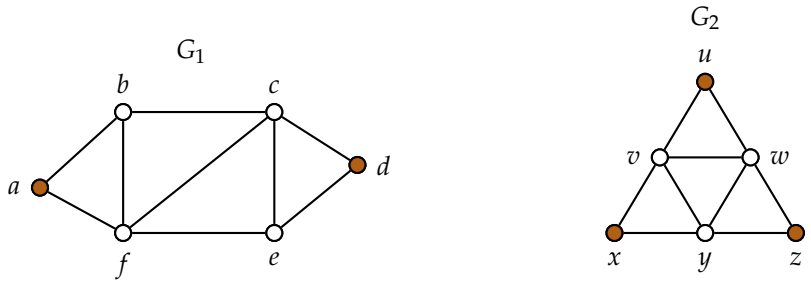


Figure 11. The example from lecture: 6 vertices and 9 edges each. Orange: the vertices adjacent to exactly 2 other vertices. G_1 has two of them, G_2 has three.

Simple example: same counts, different graphs

The path $1,2,3,4$ (Figure 8) and the "star" with centre 0 joined to $1,2,3$ both have 4 vertices and 3 edges and are both connected. But the star has a vertex adjacent to 3 others and the path does not. An isomorphism would have to send the centre to a vertex with 3 neighbours, so they are **not** isomorphic.

Comprehensive example: G_1 and G_2 of Figure 11 are not isomorphic

Solution (what you write)	**Commentary (what it does)**
Statement. Each of these graphs has 6 vertices and 9 edges. They are **not** isomorphic.	*Equal counts of vertices and edges do not settle the question.*
Solution. In G_1, vertex a is adjacent to 2 other vertices (b and f). Therefore, if there exists a graph isomorphism $f\colon V_1 \to V_2$, then $f(a)$ must also be adjacent to exactly 2 vertices, so $f(a)$ is one of u, x or z; say $f(a) = u$.	*An isomorphism preserves the number of neighbours: the neighbours of a go one-to-one onto the neighbours of $f(a)$.*
A similar situation exists for d (adjacent to c and e only), so $f(d)$ is one of the remaining x, z; say $f(d) = x$.	*The choices made with "say" do not matter, as the next step shows.*
In graph G_2 there remains one vertex, z, that is adjacent to 2 other vertices, and there is no other such vertex in G_1 (every other vertex of G_1 has 3 or 4 neighbours). So z cannot be f of anything, and f is not onto. Consequently, these graphs are **not isomorphic**. ■	*Shorter form: G_1 has **two** vertices with 2 neighbours, G_2 has **three**. This count is an invariant.*

Invariant (same in isomorphic graphs)	**Where it was used**
number of vertices, number of edges	first thing to check
connected or disconnected	theorem in Section 2.3
number of cycles of a given length	finding f in Figure 10
how many vertices have exactly k neighbours (degrees)	Figure 11; made precise in Section 3

Table 2. Some invariants. If any one of them differs, the graphs are not isomorphic. If all of them agree, the graphs may still fail to be isomorphic; to prove they are, give f.

Non-isomorphism in one breath

- Find an invariant that differs; that is a complete proof.
- Counting neighbours is the most useful quick test. It leads to the notion of degree.

3 Vertex degree (beginning of Section 11.3)

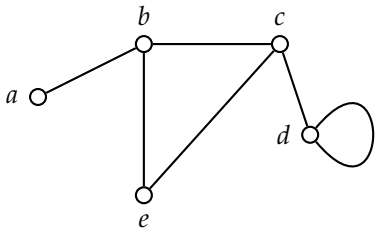
The intuition

In the last example the **number of edges incident with a vertex** was used to show that two graphs are not isomorphic. This number is called the **degree** of the vertex. In a directed graph we separate it into the arcs coming **in** and the arcs going **out**.

Degree (undirected)

Let G be an undirected graph or **multigraph** (a graph in which several edges may join the same pair of vertices). For each vertex v of G, the **degree of** v, written $\deg(v)$, is the number of edges

in G that are incident with v. **A loop is considered as 2 incident edges for v.**



$\deg(a) = 1,\ \deg(b) = 3,\ \deg(c) = 3,$
$\deg(d) = 3,\ \deg(e) = 2$

Figure 12. The example from lecture. The edge at d is a loop and is counted twice: $\deg(d) = 1 + 2 = 3$.

Simple example: degrees in K_n and in Figure 11

In K_n every vertex is adjacent to the other $n-1$, so every vertex has degree $n-1$. In Figure 11 the degrees are: G_1: a:2, b:3, c:4, d:2, e:3, f:4; G_2: u:2, v:4, w:4, x:2, y:4, z:2. The lists differ (two 2's against three), which is the argument of Section 2.4 in one line.

In-degree and out-degree

Let $G = (V, E)$ be a directed graph or multigraph. For each $v \in V$:
(a) the **incoming**, or **in-degree**, of v is the number of edges in G that are incident **into** v; it is denoted $\mathrm{id}(v)$;
(b) the **outgoing**, or **out-degree**, of v is the number of edges in G that are incident **from** v; it is denoted $\mathrm{od}(v)$.
A loop is counted once in the in-degree and once in the out-degree.

Comprehensive example: the digraph H of Figure 5, with a loop added

H has arcs $(x,y), (x,u), (v,y), (v,u), (u,v), (u,y)$. Count, for each vertex, the arcs pointing in and the arcs pointing out:

	x	y	u	v	total
id	0	3	2	1	6
od	2	0	2	2	6

Now add a loop (x,x). It adds 1 to $\mathrm{id}(x)$ and 1 to $\mathrm{od}(x)$, so $\mathrm{id}(x) = 1$, $\mathrm{od}(x) = 3$, and both totals become 7.
Observe. Each arc has exactly one tail and one head, so in both versions $\sum_v \mathrm{id}(v) = \sum_v \mathrm{od}(v) = |E|$. Similarly, in Figure 12 the degrees add up to $1+3+3+3+2 = 12 = 2 \cdot 6$, twice the number of edges. These observations are the first theorem of Section 11.3, coming next.

Counting loops

Undirected: a loop adds 2 to the degree of its vertex. Directed: a loop adds 1 to the in-degree **and** 1 to the out-degree. Counting a loop as 1 in an undirected graph is the most common slip in this section.

Degrees in one breath

- $\deg(v)$: edges incident with v, loops counted twice.
- $\mathrm{id}(v)$: arcs into v; $\mathrm{od}(v)$: arcs out of v; a loop counts once in each.

4 Templates, mistakes and exercises

The lecture at a glance

Notion	Key fact
digraph / graph	$E \subseteq V \times V$ / edges are sets $\{a, b\}$
trail / path	no repeated edge / no repeated vertex; closed versions: circuit / cycle
connected	path between any two distinct vertices; a trail can be shortened to a path
subgraph / spanning	$V_1 \subseteq V$, $E_1 \subseteq E$, edges inside V_1 / in addition $V_1 = V$
K_n / $\overline{G}$	all $\binom{n}{2}$ edges / the edges of K_n not in G
isomorphism	bijection f with $\{a, b\} \in E_1 \iff \{f(a), f(b)\} \in E_2$
degree	$\deg(v)$, loop counts 2; $\mathrm{id}(v)$, $\mathrm{od}(v)$, loop counts 1 each

4.1 Writing templates

Writing template: proving that two graphs are isomorphic

Statement. *["G_1 and G_2 are isomorphic."]*
Solution. *[Define $f : V_1 \to V_2$ vertex by vertex. Say why f is one-to-one and onto. List every edge $\{a, b\}$ of G_1 with its image $\{f(a), f(b)\}$ and confirm it is an edge of G_2. Note that $|E_1| = |E_2|$, so non-edges go to non-edges.]*
Conclusion. *["Hence f is a graph isomorphism and $G_1 \cong G_2$." ■]*

Writing template: proving that two graphs are not isomorphic

Statement. *["G_1 and G_2 are not isomorphic."]*
Solution. *[Name an invariant (number of vertices or edges, connectivity, number of vertices of degree k, number of cycles of length k, ...). Compute it for both graphs and show the values differ.]*
Conclusion. *["An isomorphism would preserve this property, so no isomorphism exists." ■]*

Phrasebook

Where	What to write
Trail, not path	"No edge repeats, so this is an a-b trail; it is not a path because b occurs twice."
Not a subgraph	"$\{c,d\} \in E_3$ but $\{c,d\} \notin E$, so $E_3 \not\subseteq E$."
Complement	"$\overline{G}$ has the same n vertices and the $\binom{n}{2} - \lvert E\rvert$ edges of K_n that are not in G."
Isomorphism	"Since f is one-to-one and $\lvert E_1\rvert = \lvert E_2\rvert$, the edges listed are all the edges of G_2."
Not isomorphic	"G_1 has two vertices of degree 2 and G_2 has three, so they are not isomorphic."
Degree with a loop	"The loop at d contributes 2 to $\deg(d)$."

4.2 Common mistakes

Mistakes that cost marks

1. **Path versus trail.** A trail may revisit vertices; a path may not.
2. **Using non-edges in a trail.** Every consecutive pair must be an edge of G.
3. **Subgraph with a foreign edge.** Check $E_1 \subseteq E$ edge by edge (Figure 4, G_3).
4. **Dropping isolated vertices from $\overline{G}$.** The complement has all n vertices.
5. **"Isomorphic because same number of vertices and edges."** Not enough (Figure 11).
6. **"Not isomorphic because my f failed."** One failed f proves nothing; use an invariant.
7. **Checking only one direction of (b).** Either check non-edges too, or use $\lvert E_1\rvert = \lvert E_2\rvert$ with f one-to-one.
8. **Loops.** 2 for deg, 1 each for id and od.

4.3 Exercises

E1. In Figure 2, find (a) a d-f trail that is not a path, (b) an a-e path of length 4, (c) a cycle of length 4.

E2. Draw all the spanning subgraphs of K_3. How many are there? How many spanning subgraphs does K_n have?

E3. Let G be a loop-free graph on n vertices. Show that $\deg_{\overline{G}}(v) = n - 1 - \deg_G(v)$ for every vertex v.

E4. Prove that G_1 and G_2 are isomorphic if and only if $\overline{G_1}$ and $\overline{G_2}$ are isomorphic. (Use the same f.)

E5. Is the graph G of Figure 7 isomorphic to its complement $\overline{G}$? Prove your answer.

E6. Show that the cycle on 6 vertices and two disjoint triangles are not isomorphic, although every vertex has degree 2 in both.

E7. * Prove that if G is disconnected, then $\overline{G}$ is connected. (Compare with the example of Section 2.2.)

E8. A digraph has $V = \{1,2,3,4\}$ and $E = \{(1,2),(2,3),(3,1),(3,3),(4,1),(1,4)\}$. Compute $\mathrm{id}(v)$ and $\mathrm{od}(v)$ for each v, and check that both sums equal $\lvert E\rvert$.

A Short answers to the exercises

E1. (a) d,a,b,d,c,f (repeats d). (b) a,b,c,d,e. (c) a,b,c,d,a (edges ab,bc,cd,da).

E2. A spanning subgraph is any subset of the edges: $2^3 = 8$ for K_3, and $2^{\binom{n}{2}}$ for K_n.

E3. v is adjacent in K_n to $n-1$ vertices; each of these pairs is an edge of exactly one of G, $\overline{G}$.

E4. $\{a,b\} \notin E_1 \iff \{f(a),f(b)\} \notin E_2$ is just the negation of condition (b), and the non-edges are the edges of the complements.

E5. Yes. Degrees in G: $a:3,\ b:2,\ c:1,\ d:1,\ e:3$; in $\overline{G}$: $a:1,\ b:2,\ c:3,\ d:3,\ e:1$. The map $a\to c,\ e\to d,\ b\to b,\ d\to a,\ c\to e$ works: G's edges ab, ad, ae, be, ec go to cb, ca, cd, bd, de, which are exactly the edges of $\overline{G}$.

E6. The cycle is connected, two disjoint triangles are not (connectivity is an invariant). Or: one has a cycle of length 6, the other none.

E7. Take $u \neq v$. If they are in different pieces of G, then $\{u,v\}$ is not an edge of G, so it is an edge of $\overline{G}$. If they are in the same piece, pick w in another piece: u,w,v is a path in $\overline{G}$.

E8. id: $1:2,\ 2:1,\ 3:2,\ 4:1$ (sum 6). od: $1:2,\ 2:1,\ 3:2,\ 4:1$ (sum 6). The loop $(3,3)$ counts once in each.