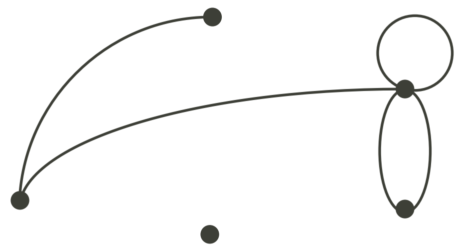
Chapter 1

Definitions and Fundamental Concepts

1.1 Definitions

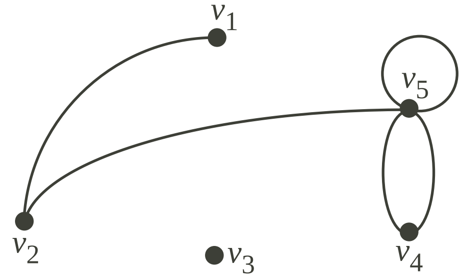
Conceptually, a *graph* is formed by *vertices* and *edges* connecting the vertices.

Example.



Formally, a graph is a pair of sets (V, E), where V is the *set of vertices* and E is the *set of edges*, formed by pairs of vertices. E is a *multiset*, in other words, its elements can occur more than once so that every element has a *multiplicity*. Often, we label the vertices with letters (for example: $a, b, c, \ldots$ or $v_1, v_2, \ldots$) or numbers $1, 2, \ldots$ Throughout this lecture material, we will label the elements of V in this way.

Example. *(Continuing from the previous example) We label the vertices as follows:*

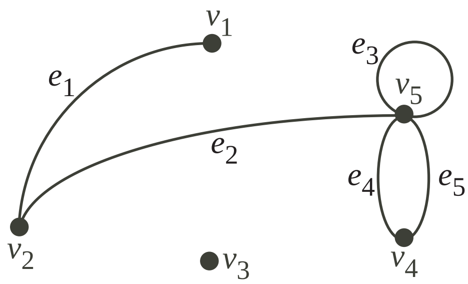


We have $V = \{v_1, \ldots, v_5\}$ *for the vertices and* $E = \{(v_1, v_2), (v_2, v_5), (v_5, v_5), (v_5, v_4), (v_5, v_4)\}$ *for the edges.*

Similarly, we often label the edges with letters (for example: $a, b, c, \ldots$ or $e_1, e_2, \ldots$) or numbers $1, 2, \ldots$ for simplicity.

Remark. *The two edges (u, v) and (v, u) are the same. In other words, the pair is not* ordered.

Example. *(Continuing from the previous example) We label the edges as follows:*



So $E = \{e_1, \ldots, e_5\}$.

We have the following terminologies:

1. The two vertices u and v are *end vertices* of the edge (u, v).
2. Edges that have the same end vertices are *parallel*.
3. An edge of the form (v, v) is a *loop*.
4. A graph is *simple* if it has no parallel edges or loops.
5. A graph with no edges (i.e. E is empty) is *empty*.
6. A graph with no vertices (i.e. V and E are empty) is a *null graph*.
7. A graph with only one vertex is *trivial*.
8. Edges are *adjacent* if they share a common end vertex.
9. Two vertices u and v are *adjacent* if they are connected by an edge, in other words, (u, v) is an edge.
10. The *degree* of the vertex v, written as $d(v)$, is the number of edges with v as an end vertex. By convention, we count a loop twice and parallel edges contribute separately.
11. A *pendant vertex* is a vertex whose degree is 1.
12. An edge that has a pendant vertex as an end vertex is a *pendant edge*.
13. An *isolated vertex* is a vertex whose degree is 0.

Example. *(Continuing from the previous example)*

- v_4 *and* v_5 *are end vertices of* e_5.
- e_4 *and* e_5 *are parallel.*
- e_3 *is a loop.*
- *The graph is not simple.*
- e_1 *and* e_2 *are adjacent.*

- v_1 *and* v_2 *are adjacent.*
- *The degree of* v_1 *is* 1 *so it is a pendant vertex.*
- e_1 *is a pendant edge.*
- *The degree of* v_5 *is* 5.
- *The degree of* v_4 *is* 2.
- *The degree of* v_3 *is* 0 *so it is an isolated vertex.*

In the future, we will label graphs with letters, for example:

$$G = (V, E).$$

The *minimum degree* of the vertices in a graph G is denoted $\delta(G)$ ($= 0$ if there is an isolated vertex in G). Similarly, we write $\Delta(G)$ as the *maximum degree* of vertices in G.

Example. *(Continuing from the previous example)* $\delta(G) = 0$ *and* $\Delta(G) = 5$.

Remark. *In this course, we only consider* finite *graphs, i.e.* V *and* E *are finite sets.*

Since every edge has two end vertices, we get

Theorem 1.1. *The graph* $G = (V, E)$*, where* $V = \{v_1, \ldots, v_n\}$ *and* $E = \{e_1, \ldots, e_m\}$*, satisfies*

$$\sum_{i=1}^{n} d(v_i) = 2m.$$

Corollary. *Every graph has an even number of vertices of odd degree.*

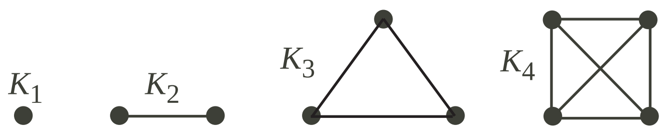
Proof. If the vertices $v_1, \ldots, v_k$ have odd degrees and the vertices $v_{k+1}, \ldots, v_n$ have even degrees, then (Theorem 1.1)

$$d(v_1) + \cdots + d(v_k) = 2m - d(v_{k+1}) - \cdots - d(v_n)$$

is even. Therefore, k is even. $\square$

Example. *(Continuing from the previous example) Now the sum of the degrees is* $1 + 2 + 0 + 2 + 5 = 10 = 2 \cdot 5$. *There are two vertices of odd degree, namely* v_1 *and* v_5.

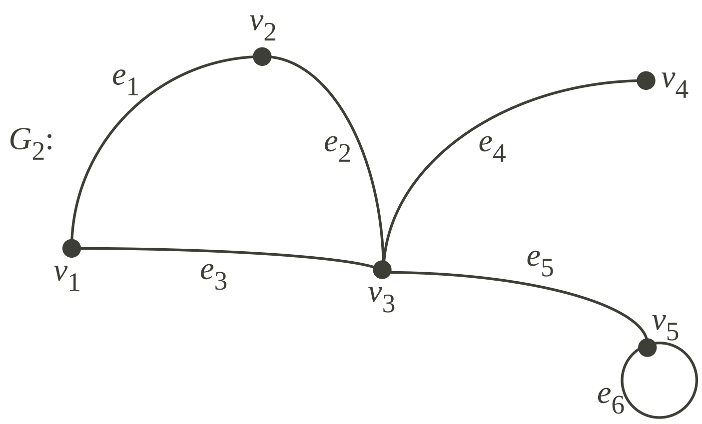
A simple graph that contains every possible edge between all the vertices is called a *complete graph*. A complete graph with n vertices is denoted as K_n. The first four complete graphs are given as examples:



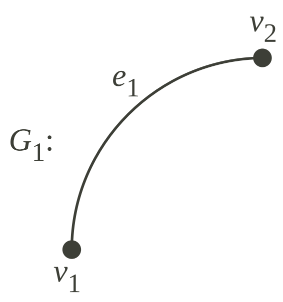
The graph $G_1 = (V_1, E_1)$ is a *subgraph* of $G_2 = (V_2, E_2)$ if

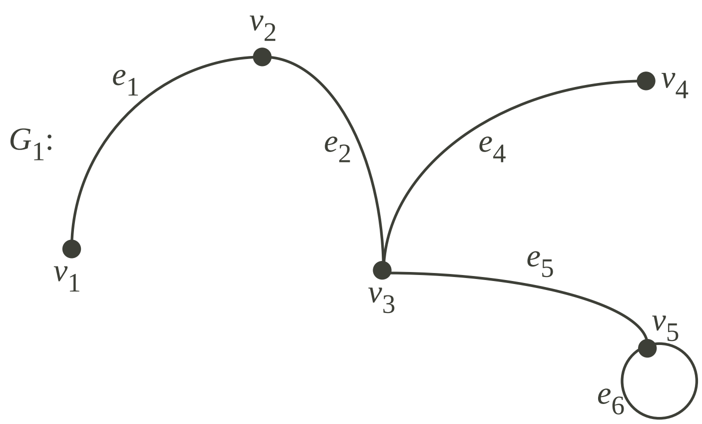
1. $V_1 \subseteq V_2$ and
2. Every edge of G_1 is also an edge of G_2.

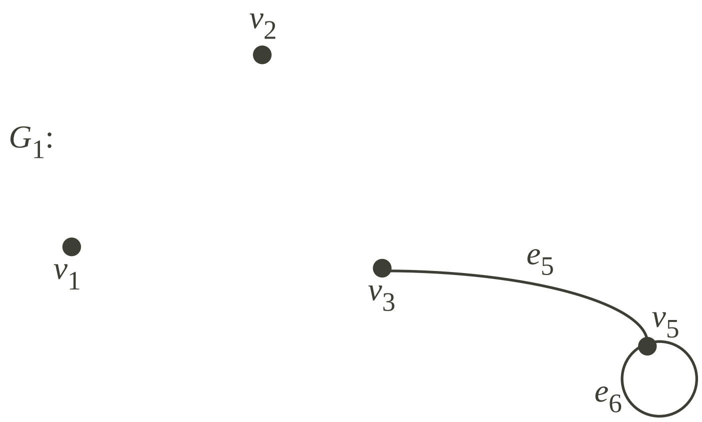
Example. *We have the graph*



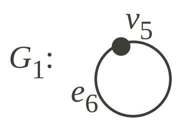
and some of its subgraphs are







and

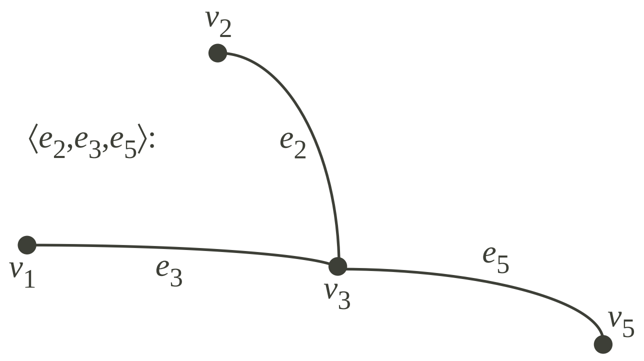


The *subgraph of* $G = (V, E)$ *induced by the edge set* $E_1 \subseteq E$ is:

$$G_1 = (V_1, E_1) =_{\text{def.}} \langle E_1 \rangle,$$

where V_1 consists of every end vertex of the edges in E_1.

Example. *(Continuing from above) From the original graph* G*, the edges* e_2*,* e_3 *and* e_5 *induce the subgraph*

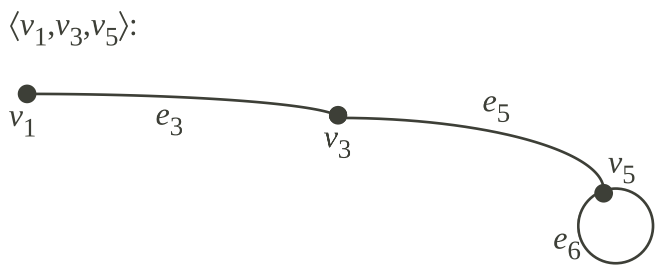


The *subgraph of* $G = (V, E)$ *induced by the vertex set* $V_1 \subseteq V$ is:

$$G_1 = (V_1, E_1) =_{\text{def.}} \langle V_1 \rangle,$$

where E_1 consists of every edge between the vertices in V_1.

Example. *(Continuing from the previous example) From the original graph* G*, the vertices* v_1*,* v_3 *and* v_5 *induce the subgraph*



A complete subgraph of G is called a *clique* of G.

1.2 Walks, Trails, Paths, Circuits, Connectivity, Components

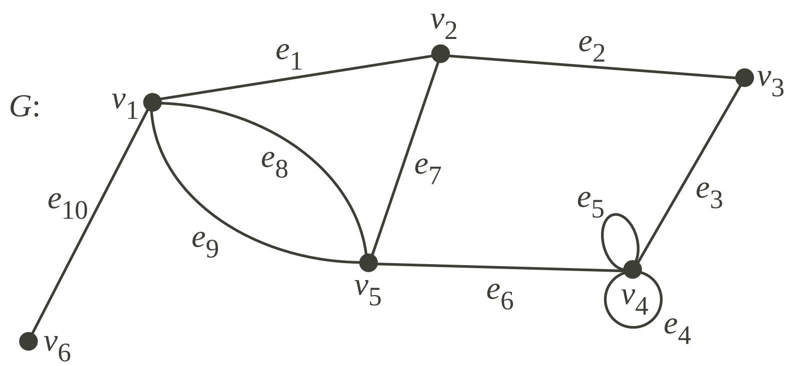
Remark. *There are many different variations of the following terminologies. We will adhere to the definitions given here.*

A *walk* in the graph $G = (V, E)$ is a finite sequence of the form

$$v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{i_k},$$

which consists of alternating vertices and edges of G. The walk starts at a vertex. Vertices $v_{i_{t-1}}$ and v_{i_t} are end vertices of e_{j_t} ($t = 1, \ldots, k$). v_{i_0} is the *initial vertex* and v_{i_k} is the *terminal vertex*. k is the *length* of the walk. A zero length walk is just a single vertex v_{i_0}. It is allowed to visit a vertex or go through an edge more than once. A walk is *open* if $v_{i_0} \neq v_{i_k}$. Otherwise it is *closed*.

Example. *In the graph*



the walk

$$v_2, e_7, v_5, e_8, v_1, e_8, v_5, e_6, v_4, e_5, v_4, e_5, v_4$$

is open. On the other hand, the walk

$$v_4, e_5, v_4, e_3, v_3, e_2, v_2, e_7, v_5, e_6, v_4$$

is closed.

A walk is a *trail* if any edge is traversed at most once. Then, the number of times that the vertex pair u, v can appear as consecutive vertices in a trail is at most the number of parallel edges connecting u and v.

Example. *(Continuing from the previous example) The walk in the graph*

$$v_1, e_8, v_5, e_9, v_1, e_1, v_2, e_7, v_5, e_6, v_4, e_5, v_4, e_4, v_4$$

is a trail.

A trail is a *path* if any vertex is visited at most once except possibly the initial and terminal vertices when they are the same. A closed path is a *circuit*. For simplicity, we will assume in the future that a circuit is not empty, i.e. its length ≥ 1. We identify the paths and circuits with the subgraphs induced by their edges.

Example. *(Continuing from the previous example) The walk*

$$v_2, e_7, v_5, e_6, v_4, e_3, v_3$$

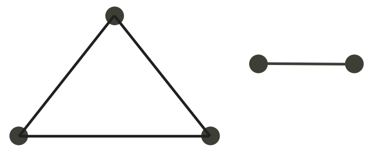
is a path and the walk

$$v_2, e_7, v_5, e_6, v_4, e_3, v_3, e_2, v_2$$

is a circuit.

The walk starting at u and ending at v is called an u–v *walk*. u and v are *connected* if there is a u–v walk in the graph (then there is also a u–v path!). If u and v are connected and v and w are connected, then u and w are also connected, i.e. if there is a u–v walk and a v–w walk, then there is also a u–w walk. A graph is *connected* if all the vertices are connected to each other. (A trivial graph is connected by convention.)

Example. *The graph*



is not connected.

The subgraph G_1 (not a null graph) of the graph G is a *component* of G if

1. G_1 is connected and
2. Either G_1 is trivial (one single isolated vertex of G) or G_1 is not trivial and G_1 is the subgraph induced by those edges of G that have one end vertex in G_1.

Different components of the same graph do not have any common vertices because of the following theorem.

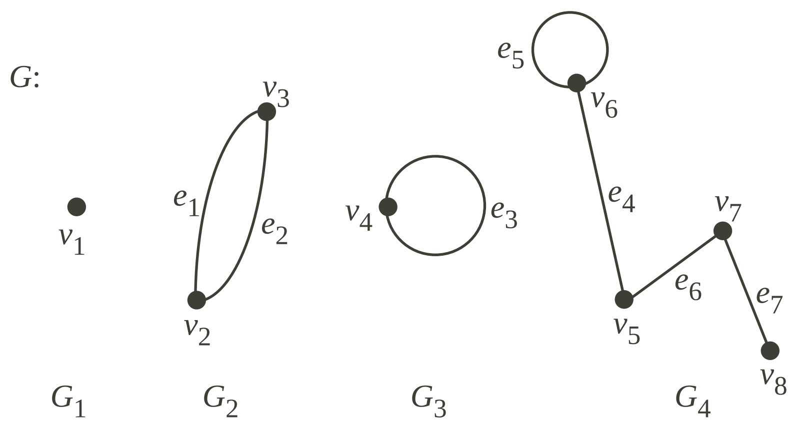
Theorem 1.2. *If the graph G has a vertex v that is connected to a vertex of the component G_1 of G, then v is also a vertex of G_1.*

Proof. If v is connected to vertex v' of G_1, then there is a walk in G

$$v = v_{i_0}, e_{j_1}, v_{i_1}, \ldots, v_{i_{k-1}}, e_{j_k}, v_{i_k} = v'.$$

Since v' is a vertex of G_1, then (condition #2 above) e_{j_k} is an edge of G_1 and $v_{i_{k-1}}$ is a vertex of G_1. We continue this process and see that v is a vertex of G_1. □

Example.



The components of G are G_1, G_2, G_3 and G_4.

Theorem 1.3. *Every vertex of G belongs to exactly one component of G. Similarly, every edge of G belongs to exactly one component of G.*

Proof. We choose a vertex v in G. We do the following as many times as possible starting with $V_1 = \{v\}$:

$(*)$ If v' is a vertex of G such that $v' \notin V_1$ and v' is connected to some vertex of V_1, then $V_1 \leftarrow V_1 \cup \{v'\}$.

Since there is a finite number of vertices in G, the process stops eventually. The last V_1 induces a subgraph G_1 of G that is the component of G containing v. G_1 is connected because its vertices are connected to v so they are also connected to each other. Condition #2 holds because we can not repeat $(*)$. By Theorem 1.2, v does not belong to any other component.

The edges of the graph are incident to the end vertices of the components. □

Theorem 1.3 divides a graph into distinct components. The proof of the theorem gives an algorithm to do that. We have to repeat what we did in the proof as long as we have free vertices that do not belong to any component. Every isolated vertex forms its own component. A connected graph has only one component, namely, itself.

A graph G with n vertices, m edges and k components has the *rank*

$$\rho(G) = n - k.$$

The *nullity* of the graph is

$$\mu(G) = m - n + k.$$

We see that $\rho(G) \geq 0$ and $\rho(G) + \mu(G) = m$. In addition, $\mu(G) \geq 0$ because

Theorem 1.4. $\rho(G) \leq m$

Proof. We will use the second principle of induction (strong induction) for m.

Induction Basis: $m = 0$. The components are trivial and $n = k$.

Induction Hypothesis: The theorem is true for $m < p$. ($p \geq 1$)

Induction Statement: The theorem is true for $m = p$.

Induction Statement Proof: We choose a component G_1 of G which has at least one edge. We label that edge e and the end vertices u and v. We also label G_2 as the subgraph of G and G_1, obtained by removing the edge e from G_1 (but not the vertices u and v). We label G' as the graph obtained by removing the edge e from G (but not the vertices u and v) and let k' be the number of components of G'. We have two cases:

1. G_2 is connected. Then, $k' = k$. We use the Induction Hypothesis on G':

$$n - k = n - k' = \rho(G') \leq m - 1 < m.$$

2. G_2 is not connected. Then there is only one path between u and v:

$$u, e, v$$

and no other path. Thus, there are two components in G_2 and $k' = k + 1$. We use the Induction Hypothesis on G':

$$\rho(G') = n - k' = n - k - 1 \leq m - 1.$$

Hence $n - k \leq m$. □

These kind of combinatorial results have many consequences. For example:

Theorem 1.5. *If G is a connected graph and $k \geq 2$ is the maximum path length, then any two paths in G with length k share at least one common vertex.*

Proof. We only consider the case where the paths are not circuits (Other cases can be proven in a similar way.). Consider two paths of G with length k:

$$v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{i_k} \quad \text{(path } p_1\text{)}$$

and

$$v_{i'_0}, e_{j'_1}, v_{i'_1}, e_{j'_2}, \ldots, e_{j'_k}, v_{i'_k} \quad \text{(path } p_2\text{).}$$

Let us consider the counter hypothesis: The paths p_1 and p_2 do not share a common vertex. Since G is connected, there exists an v_{i_0}–$v_{i'_k}$ path. We then find the last vertex on this path which is also on p_1 (at least v_{i_0} is on p_1) and we label that vertex v_{i_t}. We find the first vertex of the v_{i_t}–$v_{i'_k}$ path which is also on p_2 (at least $v_{i'_k}$ is on p_2) and we label that vertex $v_{i'_s}$. So we get a v_{i_t}–$v_{i'_s}$ path

$$v_{i_t}, e_{j''_1}, \ldots, e_{j''_\ell}, v_{i'_s}.$$

The situation is as follows:

$$\begin{array}{c}
v_{i_0}, e_{j_1}, v_{i_1}, \ldots, v_{i_t}, e_{j_{t+1}}, \ldots, e_{j_k}, v_{i_k} \\
e_{j''_1} \\
\vdots \\
e_{j''_\ell} \\
v_{i'_0}, e_{j'_1}, v_{i'_1}, \ldots, v_{i'_s}, e_{j'_{s+1}}, \ldots, e_{j'_k}, v_{i'_k}
\end{array}$$

From here we get two paths: v_{i_0}–$v_{i'_k}$ path and $v_{i'_0}$–v_{i_k} path. The two cases are:

- $t \geq s$: Now the length of the v_{i_0}–$v_{i'_k}$ path is $\geq k + 1$. $\surd$[1]
- $t < s$: Now the length of the $v_{i'_0}$–v_{i_k} path is $\geq k + 1$. $\surd$ □

A graph is *circuitless* if it does not have any circuit in it.

Theorem 1.6. *A graph is circuitless exactly when there are no loops and there is at most one path between any two given vertices.*

Proof. First let us assume G is circuitless. Then, there are no loops in G. Let us assume the counter hypothesis: There are two different paths between distinct vertices u and v in G:

$$u = v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{i_k} = v \quad \text{(path } p_1\text{)}$$

and

$$u = v_{i'_0}, e_{j'_1}, v_{i'_1}, e_{j'_2}, \ldots, e_{j'_\ell}, v_{i'_\ell} = v \quad \text{(path } p_2\text{)}$$

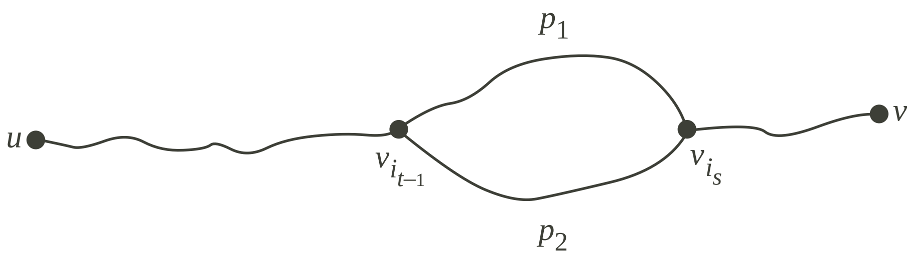
(here we have $i_0 = i'_0$ and $i_k = i'_\ell$), where $k \geq \ell$. We choose the smallest index t such that

$$v_{i_t} \neq v_{i'_t}.$$

There is such a t because otherwise

[1] From now on, the symbol $\surd$ means contradiction. If we get a contradiction by proceeding from the assumptions, the hypothesis must be wrong.

1. $k > \ell$ and $v_{i_k} = v = v_{i'_\ell} = v_{i_\ell}$ ($\checkmark$) or
2. $k = \ell$ and $v_{i_0} = v_{i'_0}, \ldots, v_{i_\ell} = v_{i'_\ell}$. Then, there would be two parallel edges between two consecutive vertices in the path. That would imply the existence of a circuit between two vertices in G. $\checkmark$



We choose the smallest index s such that $s \geq t$ and v_{i_s} is in the path p_2 (at least v_{i_k} is in p_2). We choose an index r such that $r \geq t$ and $v_{i'_r} = v_{i_s}$ (it exists because p_1 is a path). Then,

$$v_{i_{t-1}}, e_{j_t}, \ldots, e_{j_s}, v_{i_s}(= v_{i'_r}), e_{j'_r}, \ldots, e_{j'_t}, v_{i'_{t-1}}(= v_{i_{t-1}})$$

is a circuit. $\checkmark$ (Verify the case $t = s = r$.)

Let us prove the reverse implication. If the graph does not have any loops and no two distinct vertices have two different paths between them, then there is no circuit. For example, if

$$v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{i_k} = v_{i_0}$$

is a circuit, then either $k = 1$ and e_{j_1} is a loop ($\checkmark$), or $k \geq 2$ and the two vertices v_{i_0} and v_{i_1} are connected by two distinct paths

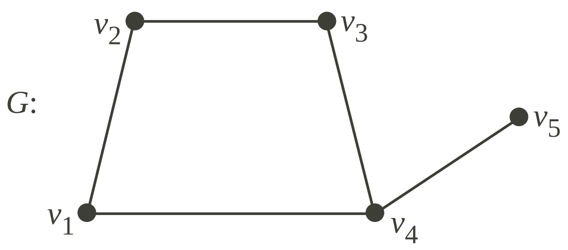
$$v_{i_0}, e_{j_1}, v_{i_1} \quad \text{and} \quad v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{i_k} = v_{i_0} \quad (\checkmark).$$

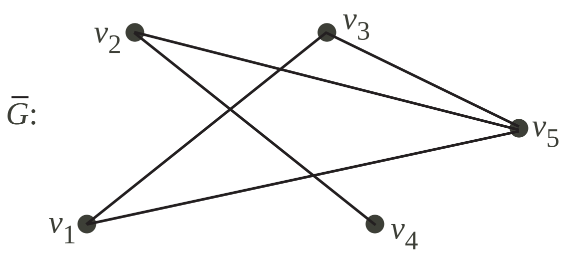
$\square$

1.3 Graph Operations

The *complement* of the simple graph $G = (V, E)$ is the simple graph $\overline{G} = (V, \overline{E})$, where the edges in $\overline{E}$ are exactly the edges not in G.

Example.

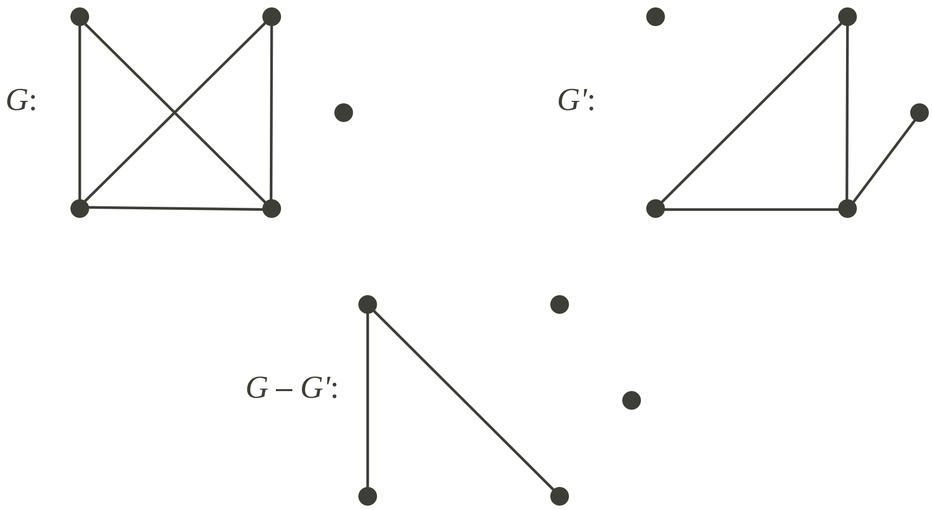




Example. *The complement of the complete graph K_n is the empty graph with n vertices.*

Obviously, $\overline{\overline{G}} = G$. If the graphs $G = (V, E)$ and $G' = (V', E')$ are simple and $V' \subseteq V$, then the *difference* graph is $G - G' = (V, E'')$, where E'' contains those edges from G that are not in G' (simple graph).

Example.



Here are some binary operations between two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$:

- The *union* is $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ (simple graph).
- The *intersection* is $G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ (simple graph).
- The *ring sum* $G_1 \oplus G_2$ is the subgraph of $G_1 \cup G_2$ induced by the edge set $E_1 \oplus E_2$ (simple graph). *Note!* The set operation $\oplus$ is the *symmetric difference*, i.e.

$$E_1 \oplus E_2 = (E_1 - E_2) \cup (E_2 - E_1).$$

Since the ring sum is a subgraph induced by an edge set, there are no isolated vertices. All three operations are commutative and associative.

Example. *For the graphs*