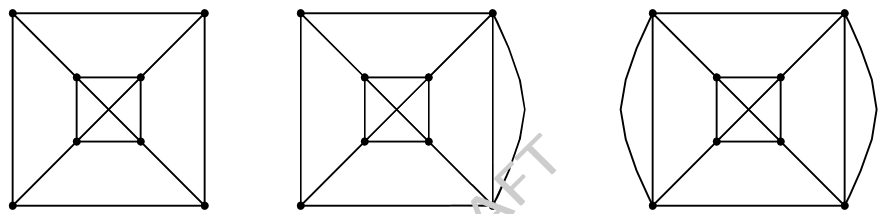
MODULE 5

Graphs - I

Basic concepts

<u>Experiment</u>

'Start from a dot. Move through each line exactly once. Draw it.' Which of the following pictures can be drawn? What if we want the 'starting dot to be the finishing dot'?



Later, we shall see a theorem by Euler addressing this question.

Definition . A **pseudograph** G is a pair (V, E) where V is a nonempty set and E is a multiset of 2-elements sets of points of V. The set V is called the **vertex set** and its elements are called **vertice s**. The set E is called the **edge set** and its elements are called **edges**.

Example . $G = \{1, 2, 3, 4\}, \{1, 1\}, \{1, 2\}, \{2, 2\}, \{3, 4\}, \{3, 4\}\}$ is a pseudograph.

Discussion A pseudograph can be represented in picture in the following way.

1. Put different points on the paper for vertices and label them.
2. If $\{u, v\}$ appears in E some k times, draw k distinct lines joining the points u and v.
3. A loop at u is drawn if $\{u, u\} \in E$.

Example A picture for the pseudograph in Example 9.1.2 is given in Figure 9.1.

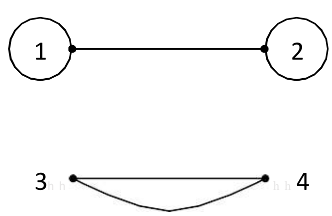


Figure 9.1: A pseudograph

Definition . Let $G = (V, E)$ be a graph. Then the following definitions and notations are in order.

1. we sometimes use $V(G)$ in place of V for the vertex set and $E(G)$ in place of E for the edge set.
2. The number $|V(G)|$ is called the **order** of the graph G, and is denoted by $|G|$. By $\|G\|$, we denote the number of edges of G. A graph with n vertices and m edges is called an (n, m) **graph**.
3. An edge $\{u, v\}$ is sometimes denoted uv. An edge uu is called a **loop**. The vertices u and v are called the **end vertices** of the edge uv. Let e be an edge. We say 'e is **incident** on u' to mean that 'u is an end vertex of e'.
4. If uv is an edge in G, then we say that the vertices u and v are **adjacent** in G, and also that u is a **neighbor** of v. We write $u \sim v$ to denote that u is adjacent to v.
5. If $v \in V(G)$, by $N(v)$ or $N_G(v)$, we denote the set of neighbors of v in G and $|N(v)|$ is called the **degree** of v. It is usually denoted by $d_G(v)$ or $d(v)$. A vertex of degree 0 is called **isolated.** A vertex of degree one is called a **pendant** vertex.
6. Two edges e_1 and e_2 are called **adjacent** if they have a common end vertex.
7. A graph is said to be **non-trivial** if it has at least one edge; else it is called a **trivial** graph.
8. A **multigraph** is a pseudograph without loops. A multigraph is a **simple graph** if no edge appears twice.
9. In this book, we consider only simple graphs with finite vertex sets. Thus, by a **graph**, we will mean a simple graph with a finite vertex set, unless stated otherwise.
10. A set of vertices or edges is said to be **independent** if no two of them are adjacent. The maximum size of an independent vertex set is called the **independence number** of G, denoted $\alpha(G)$.

Discussion Note that a graph is an algebraic structure, namely, a pair of sets satisfying some conditions. However, it is easy to describe and carry out the arguments with a pictorial representation of a graph. Henceforth, the pictorial representations are used to describe graphs and to provide our arguments, whenever required. There is no loss of generality in doing this.

Example. Consider the graph G in Figure 9.2. The vertex 12 is an isolated vertex whereas the vertex 10 is a pendant vertex. We have $N(1) = \{2, 4, 7\}$, $d(1) = 3$. The vertices 1 and 6 are not adjacent. The set $\{9, 10, 11, 2, 4, 7\}$ is an independent vertex set. The set $\{1, 2\}, \{8, 10\}, \{4, 5\}$ } is an independent edge set.

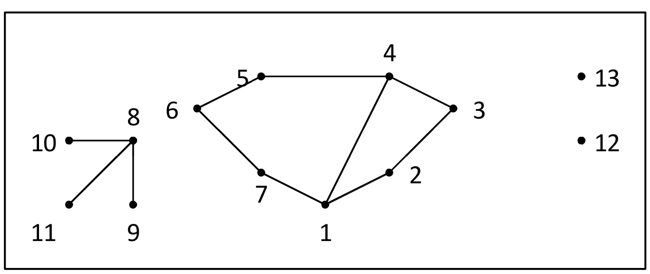


Figure 9.2: A graph G.

Definition . Let $G = (V, E)$ be a graph on n vertices, say $V = \{1, \ldots, n\}$. Then, G is said to be a

1. **Complete graph**, denoted K_n, if each pair of vertices in G are adjacent.
2. **Path graph**, denoted P_n, if $E = \{\{i, i+1\} : 1 \leq i \leq n-1\}$.
3. **Cycle graph**, denoted C_n, if $E = \{\{i, i+1\} : 1 \leq i \leq n-1\} \cup \{n, 1\}$.
4. **Bipartite graph** if $V = V_1 \cup V_2$ such that $|V_1|, |V_2| \geq 1$, $V_1 \cap V_2 = \emptyset$ and $e = \{u, v\} \in E$ if either $u \in V_1$ and $v \in V_2$, or $u \in V_2$ and $v \in V_1$.
5. **Complete bipartite graph**, denoted $K_{r,s}$ if $E = \{\{i, j\} : 1 \leq i \leq r, 1 \leq j \leq s\}$.

Figure 9.3: P_n and C_n.

The importance of the labels of the vertices depends on the context. At this point of time, even if we interchange the labels of the vertices, we still call them a complete graph or a path graph or a cycle or a complete bi-partite graph.

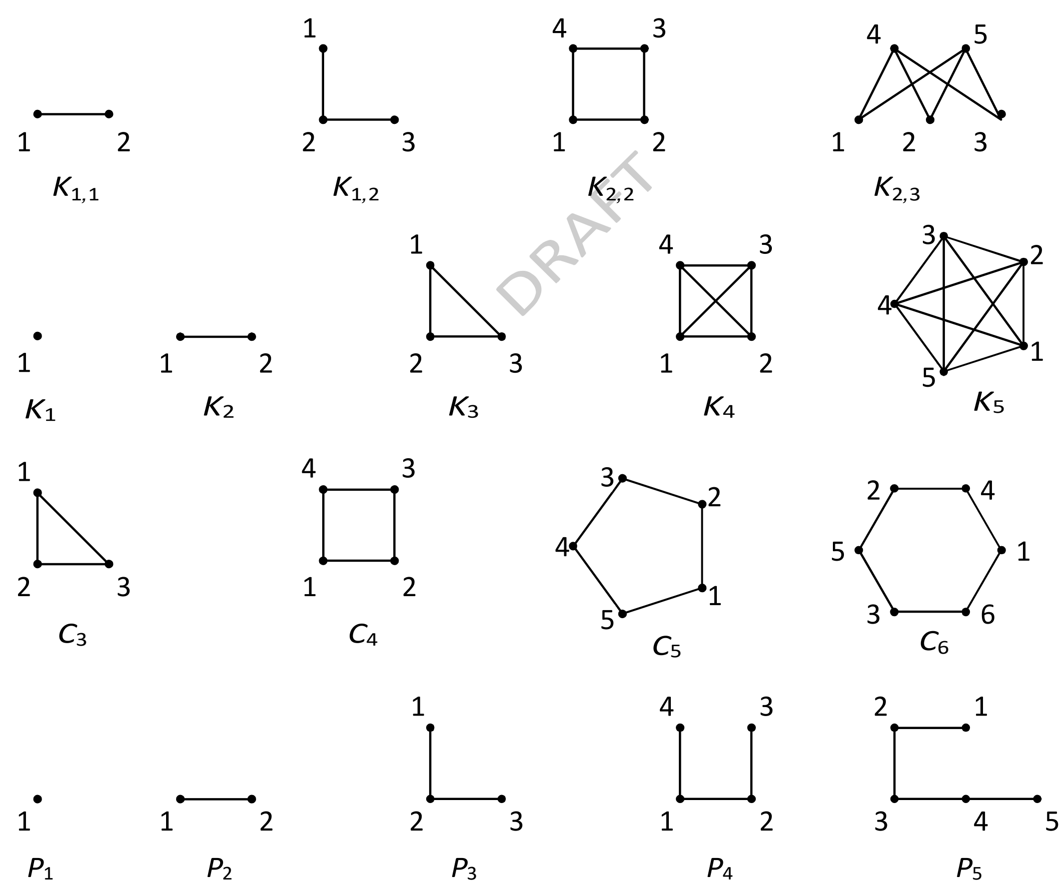


Figure : Some well known family of graphs

What is the maximum number of edges possible in a simple graph of order n?

Lemma **[Hand shaking lemma]** *In any graph (simple) G,* $\sum_{v \in V} d(v) = 2|E|$. *Thus, the number of vertices of odd degree is even.*

Proof. Each edge contributes 2 to the sum $\sum_{v \in V} d(v)$. Hence, $\sum_{v \in V} d(v) = 2|E|$. Note that

$$2|E| = \sum_{v \in V} d(v) = \sum_{v : d(v) \text{ is odd}} d(v) + \sum_{v : d(v) \text{ is even}} d(v)$$

Since $\sum_{v : d(v) \text{ is even}} d(v)$ is even, the above implies that $\sum_{v : d(v) \text{ is even}} d(v)$ must be even as well. Therefore, the number of vertices of odd degree is even. ■

In a party of 27 persons, prove that someone must have an even number of friends assuming that friendship is mutual.

Example The graph in Figure 9.5 is called the **Petersen graph**. We shall use it as an example in many places.

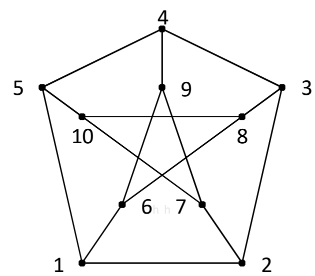


Figure : Petersen graphs

Proposition . *In a graph G with $n = |G| \geq 2$, there are two vertices of equal degree.*

Proof. If G has two or more isolated vertices, we are done. First, suppose G has exactly one isolated vertex. Then, the remaining $n-1$ vertices have degrees between 1 and $n-2$ and hence by PHP, the result follows. Otherwise, G has no isolated vertex. Then G has n vertices whose degrees lie between 1 and $n-1$. Again by PHP, we get the required result. ■

EXERCISE

1. *Let $G = (V, E)$ be a graph with a vertex $v \in V$ of odd degree. Then, prove that there exists a vertex $u \in V$ such that there is a path from v to u and* $\deg(u)$ *is also odd.*
2. *Let $G = (V, E)$ be a graph having exactly two vertices, say u and v, of odd degree. Then, prove that there is a path in G connecting u and v.*

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

1. the **minimum degree** of a vertex in G is denoted by $\delta(G)$ and the **maximum degree** of a vertex in G is denoted by $\Delta(G)$.
2. a graph G is called ***k*-regular** if $d(v) = k$ for all $v \in V(G)$.
3. a 3-regular graph is called **cubic**.

Example

1. The cycle graph C_n is 2-regular whereas the complete graph K_n is $(n-1)$-regular.
2. The Petersen graph and the complete graph K_4 are cubic.
3. The graph P_4 is not regular.
4. Consider the graph G in Figure 9.2. We have $\delta(G) = 0$ and $\Delta(G) = 3$.

Can we have a cubic graph on 5 *vertices?*

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

1. Then a graph H is a called a **subgraph** of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.
2. Then a subgraph H of G is called a **spanning subgraph** if $V(G) = V(H)$.
3. Then a k-regular spanning subgraph is called a k-**factor** of G.
4. If $U \subseteq V(G)$, then the **induced subgraph** of G on U is denoted by $\langle U \rangle = (U, E)$, where the edge set $E = \{\{u, v\} \in E(G) : u, v \in U\}$.

Example . Consider the graph G in Figure 9.2.

(a) Let H_1 be the graph with $V(H_1) = \{6, 7, 8, 9, 10, 12\}$ and $E(H_1) = \{\{6, 7\}, \{9, 10\}\}$. Then, H_1 is not a subgraph of G as $\{9, 10\} \notin E(G)$.

(b) Let H_2 be the graph with $V(H_2) = \{6, 7, 8, 9, 10, 12\}$ and $E(H_2) = \{\{6, 7\}, \{8, 10\}\}$. Then, H_2 is a subgraph but not an induced subgraph of G as $\{8, 9\} \in E(G)$ but not in $E(H_2)$.

(c) Let H_3 be the induced subgraph of G on the vertex set $\{6, 7, 8, 9, 10, 12\}$. Then, verify that $E(H_3) = \{\{6, 7\}, \{8, 9\}, \{8, 10\}\}$.

(d) The graph G does not have a 1-factor.

2. A complete graph has a 1-factor if and only if it has an even order.
3. The Petersen graph has many 1-factors. One of them is obtained by selecting the edges $\{1, 6\}, \{2, 7\}, \{3, 8\}, \{4, 9\}$ and $\{5, 10\}$.

Consider K_8 on the vertex set $\{1, 2, \ldots, 8\}$. How many 1-factors does it have?

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

1. If $v \in V(G)$ then the graph $G - v$, called the **vertex deleted subgraph**, is obtained from G by deleting v and all the edges that are incident with v.
2. If $e \in E(G)$, then the graph $G - e = (V, E(G) \setminus \{e\})$ is called the **edge deleted subgraph**.
3. If $u, v \in V(G)$ such that $u \nsim v$, then $G + uv = (V, E(G) \cup \{uv\})$ is called the **graph obtained by edge addition**.
4. The **complement** $\overline{G}$ of a graph G is defined as $(V(G), E)$, where $E = \{uv : u \neq v, uv \notin E(G)\}$.

Example . 1. Consider the graph G in Figure 9.2. Let H_2 be the graph with $V(H_2) = \{6, 7, 8, 9, 10, 12\}$ and $E(H_2) = \{\{6, 7\}, \{8, 10\}\}$. Consider the edge $e = \{8, 9\}$. Then, $H_2 + e$ is the induced subgraph $\langle\{6, 7, 8, 9, 10, 12\}\rangle$ and $H_2 - 8 = \langle\{6, 7, 9, 10, 12\}\rangle$.

2. See Figure 9.6 for two examples of complement graphs.

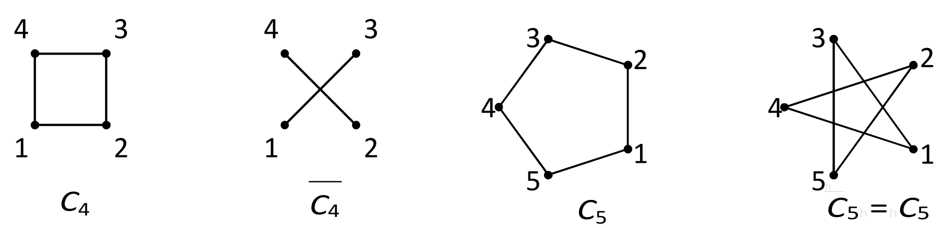


Figure: Complement graphs

3. The complement of K_3 contains 3 isolated points/vertices.
4. For any graph G, $\|G\| + \|\overline{G}\| = \binom{|G|}{2}$.
5. In any graph G of order n, $d_G(v) + d_{\overline{G}}(v) = n - 1$. Thus, $\Delta(G) + \Delta(\overline{G}) \geq n - 1$.

t $\Delta(G) + \Delta(\overline{G}) = n - 1$. 1. *Characterize graphs G such tha*

2. *Can we have a graph G such that* $\Delta(G) + \Delta(\overline{G}) = n$?
3. *Show that a k-regular simple graph on n vertices exists if and only if kn is even and* $n \geq k + 1$ Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two graphs.

1. Then their **intersection**, denoted $G \cap H$, is defined as $(V(G) \cap V(H), E(G) \cap E(H))$.
2. Then their **union**, denoted $G \cup H$, is defined as $(V(G) \cup V(H), E(G) \cup E(H))$.
3. Then their **disjoint union** is the union while treating the vertex sets as disjoint sets.
4. If $V(G) \cap V(H) = \emptyset$, then their **join**, denoted $G + H$ has $V(G) \cup V(H)$ as the vertex set and $E(G) \cup E(H) \cup \{uv : u \in V(G), v \in V(H)\}$ as the edge set.
5. Then their **Cartesian product**, denoted $G \times H$, has $V(G) \times V(H)$ as the vertex set and the edg e set consists of all elements $\{(u, v), (u', v')\}$, where either $u = u'$ and $\{v, v'\} \in E(H)$, or $v = v'$ and $\{u, u'\} \in E(G)$.

Example . Two graphs G and H with their intersection $G \cap H$, their union $G \cup H$ and their disjoin t union as G_1 are shown in Figure 9.7. Further, the join of $K_2 + K_3$ and $K_2 + \overline{K_2}$ are also given. Note that $K_2 + K_3 = K_5$ and $\overline{K_2} + \overline{K_2} = C_4$.

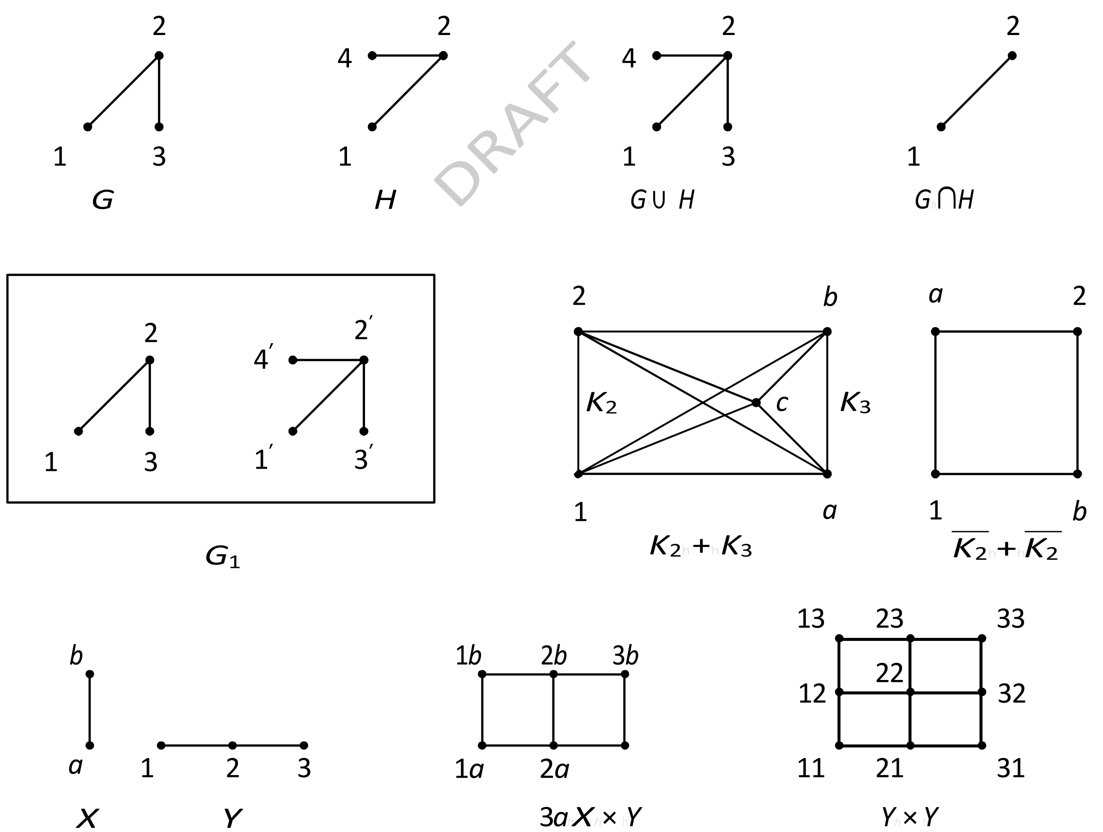


Figure : Examples of graph constructions (see Definition 9.1.24)

1. *What is the complement of the disjoint union of* $\overline{G}$ *and* $\overline{H}$?
2. *Is* $K_{m,n} = \overline{K_m} + \overline{K_n}$?

Connectedness

Definition Let $G = (V, E)$ be a graph and let $u, v \in V$.

4. A u-v **walk** in G is a finite sequence of vertices $[u = v_1, v_2, \cdots, v_{k-1}, v_k = v]$ such that $v_i v_{i+1} \in E$, for all $i = 1, \cdots, k-1$.
5. The **length** of a walk is the number of edges on it.
6. A walk is called a **trail** if edges on the walk are not repeated.
7. A u-v walk is a called a **path** if the vertices involved are all distinct, except that u and v can be the same.
8. If P is a u-v path with $u \neq v$, then we sometimes call u and v as the **end vertices of** P and the remaining vertices on P as the **internal vertices**.
9. A walk (trail, path) is called **closed** if $u = v$.
10. The **length** of a path is the number of edges on it. A path can have length 0.
11. A closed path is called a **cycle/circuit**. Thus, in a simple graph a cycle has length at least 3. A cycle (walk, path) of length k is also written as a k-cycle (k walk, k cut-vertex).

Example 1. Take $G = K_5$ with vertex set $\{1, 2, 3, 4, 5\}$.

(a) Then [1, 2, 3, 2, 1, 2, 5, 4, 3] is an 8 walk in G and [1, 2, 2, 1] is not a walk.

(b) The walk [1, 2, 3, 4, 5, 2, 4, 1] is a closed trail.

(c) The walk [1, 2, 3, 5, 4, 1] is a closed path, *i.e.*, it is a 5-cycle.

(d) The maximum length of a cycle in G is 5 and the minimum length of a cycle in G is 3.

(e) The number of 3-cycles in G is $\binom{5}{3} = 10$.

(f) Verify that the number of 4-cycles in G is not $\binom{5}{4}$. Can it be $3 \times \binom{5}{4}$?

2. Let G be the Petersen graph. Then, G has a 9-cycle, namely, [6, 8, 10, 5, 4, 3, 2, 7, 9, 6]. But, G has no 10-cycles. We shall see this when we discuss the Hamiltonian graphs.

Proposition *Let u and v be distinct vertices in a graph G. Let $W = [u = u_1, \ldots, u_k = v]$ be a walk. Then W contains a u-v path.*

Proof. If no vertex on W repeats, then W is itself a path. So, let $u_i = u_j$ for some $i < j$. Now, consider the walk $W_1 = [u_1, \ldots, u_{i-1}, u_j, u_{j+1}, \ldots u_k]$. This is also a u-v walk but of shorter length. Thus, using induction on the length of the walk, the desired result follows. ∎

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

1. The **distance** $d(u, v)$ between two vertices $u, v \in V, u \neq v$ is the shortest length of a u-v path in G. If no such path exists, the distance is taken to be ∞.
2. The greatest distance between any two vertices in a graph G is called the **diameter** of G, and is denoted by $\text{diam}(G)$.
3. Let $dist_v = \max_{u \in V} d(v, u)$. The **radius** is the $\min_{v \in V} dist_v$ and the **center** is the set of all vertices v for which $dist_v$ is the radius.
4. The **girth**, denoted $g(G)$, of a graph G is the minimum length of a cycle contained in G. If G has no cycle, then we put $g(G) = \infty$.

Example . The Petersen graph has diameter 2, radius 2 and each vertex is in the center. Furth er, its girth is 5.

EXERCISE 1. *Determine the diameter, radius, center and girth of the following graphs:* P_n, C_n, K_n *and* $K_{n,m}$.

2. *Let* G *be a graph. Then, show that the distance function* $d(u, v)$ *is a metric on* $V(G)$. *That is, it satisfies*
   (a) $d(u, v) \geq 0$ *for all* $u, v \in V(G)$ *and* $d(u, v) = 0$ *if and only if* $u = v$,
   (b) $d(u, v) = d(v, u)$ *for all* $u, v \in V(G)$ *and*
   (c) $d(u, v) \leq d(u, w) + d(w, u)$ *for all* $u, v, w \in V(G)$.

Proposition . *Let* G *be a graph with* $\|G\| \geq 1$ *and* $d(v) \geq 2$, *for each vertex except one, say* v_1. *Then,* G *has a cycle.*

Proof. Consider a longest path $[v_1, \ldots, v_k]$ in G (as $V(G)$ is finite, such a path exists). As $d(v_k) \geq 2$, it must be adjacent to some vertex from $v_2, \ldots, v_{k-2}$; otherwise, we can extend it to a longer path. Choose $i \geq 2$ such that v_i is adjacent to v_k. Then, $[v_i, v_{i+1}, \ldots, v_k, v_i]$ is a cycle. ■

Proposition *Let* P *and* Q *be two different* u-v *paths in* G. *Then,* $P \cup Q$ *contains a cycle.*

Proof. Imagine a signal was sent from u to v via P and was returned back from v to u via Q. Call an edge 'dead' if signal has passed through it twice. Notice that each vertex receives the signal as many times as it sends the signal.

Is $E(P) = E(Q)$? No, otherwise both P and Q are the same paths.

So, there are some 'alive' edges. Get an alive edge v_1v_2. There must be an alive edge v_2v_3; otherwise, v_2 is incident to just one alive edge and some dead edges so that v_2 has received more signal than it has sent. Similarly, get v_3v_4 and so on. Stop at the first instance of repetition of a vertex: $[v_1, v_2, \cdots, v_i, v_{i+1} \cdots, v_j = v_i]$. Then, $[v_i, v_{i+1} \cdots, v_j = v_i]$ is a cycle.

Alternate. Consider the graph $H = (V(P) \cup V(Q), E(P)\Delta E(Q))$, where Δ is the symmetric difference. Notice that $E(H) \neq \emptyset$, otherwise $P = Q$. As the degree of each vertex in the multigraph $P \cup Q$ is even and H is obtained after deleting pairs of multiple edges, each vertex in H has even degree. Hence, by Proposition 9.2.7, H has a cycle. ■

Proposition *Every graph* G *containing a cycle satisfies* $g(G) \leq 2\,\mathrm{diam}(G) + 1$.

Proof. Let $C = [v_1, v_2, \ldots, v_k, v_1]$ be the shortest cycle and $\mathrm{diam}(G) = r$. If $k \geq 2r + 2$, then consider the path $P = [v_1, v_2, \ldots, v_{r+2}]$. Since the length of P is $r + 1$ and $\mathrm{diam}(G) = r$, there is a v_{r+2}-v_1 path R of length at most r. Note that P and R are different v_1-v_{r+2} paths. By Proposition 9.2.8, the closed walk $P \cup R$ of length at most $2r + 1$ contains a cycle. Hence, the length of this cycle is at most $2r + 1$, a contradiction to C having the smallest length $k \geq 2r + 2$. ■

Definition Let $C = [v_1, \ldots, v_k = v_1]$ be a cycle in a graph G. An edge v_iv_j in G is called a **chord** of C if it is not an edge of C. G is called **chordal** if each cycle of length at least 4 has a chord. G is **acyclic** if it has no cycles.

For example, complete graphs are chordal, so are the acyclic graphs. The Petersen graph is not chordal.

1. *How many acyclic graphs are there on the vertex set* $\{1, 2, 3\}$?
2. *How many chordal graphs are there on the vertex set* $\{1, 2, 3, 4\}$?

Definition 1. A graph G is said to be a **maximal** graph with respect to a property P if G has property P and no proper supergraph of G has the property P. The term **minimal** graph is defined similarly.

> Notice!
>
> The class of all graphs with that property is the poset here. So, the maximality and the minimality are defined naturally.

2. A complete subgraph of G is called a **clique**. The maximum order of a clique is called the **clique number** of G. It is denoted $\omega(G)$.
3. A graph G is called **connected** if there is a u-v path, for each $u, v \in V(G)$.
4. A graph which is not connected is called **disconnected**.

If G is a disconnected graph, then a maximal connected subgraph is called a **component** or sometimes a **connected component**.

Example Consider the graph G shown in Figure 9.2.

1. Some cliques in G are $\langle\{8, 10\}\rangle$, $\langle\{2\}\rangle$. The first is a maximal clique. Notice that every vertex is a clique. Similarly, each edge is a clique. Here $\omega(G) = 2$.
2. The graph G is not connected. It has four connected components, namely, $\langle\{8, 9, 10, 11\}\rangle$, $\langle\{1, 2, 3, 4, 5, 6, 7\}\rangle$, $\langle\{12\}\rangle$ and $\langle\{13\}\rangle$.

What is $\omega(G)$ for the Petersen graph?

Proposition . *If $\delta(G) \geq 2$, then G has a path of length $\delta(G)$ and a cycle of length at least $\delta(G) + 1$.*

Proof. Let $[v_1, \cdots, v_k]$ be a longest path in G. As $d(v_k) \geq 2$, v_k is adjacent to some vertex $v \neq v_{k-1}$. If v is not on the path, then we have a path that is longer than $[v_1, \cdots, v_k]$ path. A contradiction. So, let i be the smallest positive integer such that v_i is adjacent to v_k. Then

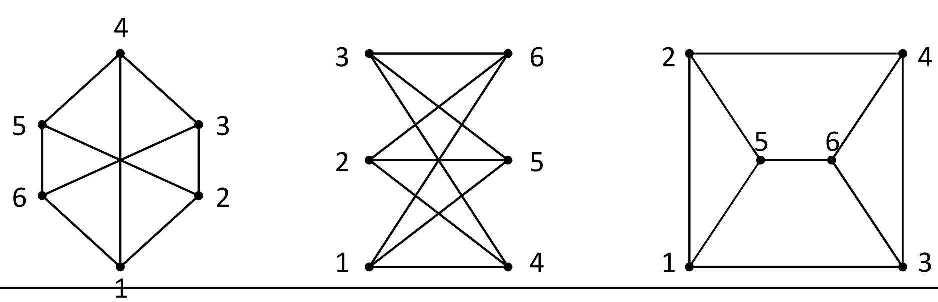
$$\delta(G) \leq d(v_k) \leq |\{v_i, v_{i+1}, \cdots, v_{k-1}\}|.$$

Hence, the cycle $C = [v_i, v_{i+1}, \cdots, v_k, v_i]$ has length at least $\delta(G) + 1$ and the length of the path $P = [v_i, v_{i+1}, \cdots, v_k]$ is at least $\delta(G)$. ■

Isomorphism in graphs

Definition Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be **isomorphic** if there is a bijection $f : V \longrightarrow V'$ such that $u \sim v$ in G if and only if $f(u) \sim f(v)$ in G', for each $u, v \in V$. In other words, an isomorphism is a bijection between the vertex sets which preserves adjacency. We write $G \cong G'$ to mean that G is isomorphic to G'.

onsider the graphs in Figure 9.8. We observe the following:



F G H

Figure 9.8: $F \cong G$ but $F \ncong H$

12. The graph F is not isomorphic to H as $\alpha(F)$, the independence number of F is 3 whereas $\alpha(H) = 2$. Alternately, H has a 3-cycle, whereas F does not have a 3-cycle.

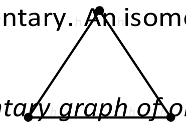
13. The map $f : V(F) \longrightarrow V(G)$ defined by $f(1) = 1$, $f(2) = 5$, $f(3) = 3$, $f(4) = 4$, $f(5) = 2$ and $f(6) = 6$ gives an isomorphism. So, $F \cong G$.

$1 \rightarrow 2, 4, 6$	$f(1) = 1 \longrightarrow f(2) = 5, f(4) = 4, f(6) = 6$
$3 \rightarrow 2, 4, 6$	$f(3) = 3 \longrightarrow f(2) = 5, f(4) = 4, f(6) = 6$
$5 \rightarrow 2, 4, 6$	$f(5) = 2 \longrightarrow f(2) = 5, f(4) = 4, f(6) = 6$

Definition . A graph G is called **self-complementary** if $G \cong \overline{G}$.

Example Let G be a self-complementary graph on n vertices. Then $\|G\| = n(n-1)/4$ as $\|G\| = \|\overline{G}\|$ and there are $\binom{n}{2}$ edges in the complete graph. Thus, either $n = 4k$ or $n = 4k + 1$. Now, verify the following:

1. The path $P_4 = [0, 1, 2, 3]$ is self complimentary. An isomorphism from P_4 to $\overline{P}_4$ is described by $f(i) = 2i \pmod 5$.
2. The cycle $C_5 = [0, 1, 2, 3, 4, 0]$ is self complimentary. An isomorphism from C_5 to $\overline{C_5}$ is describe d by $f(i) = 2i \pmod 5$.



EXERCISE *1. Construct a self-complementary graph of order 4k.*

2. Construct a self-complementary graph of order $4k + 1$.

Trees

Definition. Let G be a connected graph. A vertex v of G is called a **cut-vertex** if $G - v$ is disconnected. Thus, $G - v$ is connected if and only if v is not a cut-vertex.

Theorem *Let G be a connected graph with $|G| \geq 2$ and let $v \in V(G)$.*

14. If $d(v) = 1$, then $G - v$ is connected, so that v is never a cut-vertex.

15. If $G - v$ is connected, then either $d(v) = 1$ or v is on a cycle.

Proof. 1. Let $u, w \in V(G - v)$, $u \neq w$. As G is connected, there is a u-w path P in G. The vertex v cannot be an internal vertex of P, as each internal vertex has degree at le ast 2. Hence, the path P is available in $G - v$. So, $G - v$ is connected.

2. Assume that $G - v$ is connected. If $d_G(v) = 1$, then there is nothing to prove. So, assume that $d(v) \geq 2$. We need to show that v is on a cycle in G.

Let u and w be two distinct neighbors of v in G. As $G - v$ is connected there is a path, say $[u = u_1, \ldots, u_k = w]$, in $G - v$. Then $[u = u_1, \ldots, u_k = w, v, u]$ is a cycle in G containing v. ■

Let G be a graph and v be a vertex on a cycle. Can $G - v$ be disconnected?

Definition. Let G be a graph. An edge e in G is called a **cut-edge** or a **bridge** if $G-e$ has more connected components than that of G.

Proposition *Let G be connected and let $e = uv$ be a cut-edge. Then $G-e$ has two components, one containing u and the other containing v.*

Proof. If $G-e$ is not disconnected, then by definition, e cannot be a cut-edge. So, $G-e$ has at least two components. Let G_u (respectively, G_v) be the component containing the vertex u (respectively, v). We claim that these are the only components.

Let $w \in V(G)$. Since G is connected, there is a path, say P, from w to u. Moreover, either P contains v as its internal vertex or P does not contain v. In the first case, $w \in V(G_v)$ and in the latter case, $w \in V(G_u)$. Thus, every vertex of G is either in $V(G_v)$ or in $V(G_u)$ and hence the required result follows. ∎

Theorem *Let G be a graph and let e be an edge. Then, e is a cut-edge if and only if e is not on a cycle.*

Proof. Suppose that $e = uv$ is a cut-edge of G. Let F be the component of G that contains e. Then, by Proposition 9.4.5, $F-e$ has two components, namely, F_u that contains u and F_v that contains v. Let if possible, $C = [u, v = v_1, \ldots, v_k = u]$ be a cycle containing $e = uv$. Then $[v = v_1, \ldots, v_k = u]$ is a u-v path in $F-e$. Hence, $F-e$ is still connected. A contradiction. Thus, e cannot be on any cycle.

Conversely, let $e = uv$ be an edge which is not on any cycle. Now, suppose that F is the component of G that contains e. We need to show that $F-e$ is disconnected.

Let if possible, there is a u-v path, say $[u = u_1, \ldots, u_k = v]$, in $F-e$. Then, $[v, u = u_1, \ldots, u_k = v]$ is a cycle containing e. A contradiction to e not lying on any cycle.

Hence, e is a cut-edge of F. Consequently, e is a cut-edge of G. ∎

EXERCISE *Let G be a graph on $n > 2$ vertices. If $\|G\| > \binom{n-1}{2}$, is G necessarily connected? Give an 'if and only if' condition for the connectedness of a graph with exactly $\binom{n-1}{2}$ edges.*

Definition. A connected acyclic graph is called a **tree**. A **forest** is a graph whose components are trees.

Thus, any acyclic graph is a forest and any component of it is a tree.

Proposition *A tree on n vertices has $n-1$ edges.*

Proof. We apply strong induction on n. Take a tree on $n \geq 2$ vertices and delete an edge e. Then, we get two subtrees T_1, T_2 of order n_1, n_2, respectively, where $n_1+n_2 = n$. So, $E(T) = E(T_1) \cup E(T_2) \cup \{e\}$. By induction hypothesis $\|T\| = \|T_1\| + \|T_2\| + 1 = n_1 - 1 + n_2 - 1 + 1 = n_1 + n_2 - 1 = n - 1$.

∎

Corollary *A tree with at least two vertices has at least two pendant vertices.*

Proof. Let T be any tree on $n \geq 2$ vertices. Then $\sum_{v \in V(T)} d(v) = 2\|E(T)\| = 2(n-1) = 2n-2$. By PHP, T has at least two vertices of degree 1. ∎

Theorem. *Let G be a graph with n vertices. Then the following are equivalent:*

1. *G is a tree.*

2. *G is a maximal acyclic graph.*
3. *G is a minimal connected graph.*
4. *G is acyclic and it has $n-1$ edges.*
5. *G is connected and it has $n-1$ edges.*
6. *Between any two distinct vertices of G there exists a unique path.*

Proof. (1)e(2). Let G be a tree. On the contrary, suppose that G is not maximal acyclic. Then there exist $u, v \in V(G)$ such that $G + uv$ is acyclic. If in G, there exists a u-v path, then $G + uv$ would have a cycle containing the edge uv. So, in G, there is no u-v path. It contradicts the assumption that G is a tree and hence connected.

Conversely, suppose that G is maximal acyclic. If G is not a tree, then G has at least two components. Let u and v be two vertices from different components, so that there exists no u-v path in G. Thus $G + uv$ has no cycle. This contradicts the assumption that G is maximal acyclic.

(1)e(3). Let G be a tree. Then G is connected. Let $e = uv$ be an edge of G. By (2), e is the only u-v path. Then $G-e$ is disconnected. Hence G is minimal connected.

Conversely, suppose G is minimal connected. If G is not a tree, then there is a cycle in G. Let u, v be two adjacent vertices on such a cycle. Now, $G - uv$ is still connected. It contradicts the assumption that G is minimal connected.

(1)e(4). Let G be a tree. Then G is acyclic, and By Proposition 9.4.9, G has $n-1$ edges.

Conversely, let G by acyclic and G has $n-1$ edges. If possible, let G be disconnected. Then G has components $G_1, \ldots, G_k$, $k \geq 2$. As G is acyclic, each G_i is a tree on, say $n_i \geq 1$ vertices, with $\sum_{i=1}^{k} n_i = n$. As $k \geq 2$, we have $\|G\| = \sum_{i=1}^{k} (n_i - 1) = n - k < n - 1 = \|G\|$, a contradiction.

(1)e(5). Let G be a tree. Then G is connected, and By Proposition 9.4.9, G has $n-1$ edges.

Conversely, assume that G is connected and G has $n-1$ edges. On the contrary, suppose that G is not a tree. Then G has a cycle. Select an edge e from the cycle. Notice that $G-e$ is connected. We go on selecting edges from G that lie on cycles and keep removing them, until we get an acyclic graph H. Since the edges that are being removed lie on some cycle, the graph H is still connected. So, by definition, H is a tree on n vertices. Thus, by Proposition 9.4.9, $\|H\| = n-1$. But, in the above argument, we have deleted at least one edge and hence, $\|G\| \geq n$. This gives a contradiction to $\|G\| = n-1$.

(1)e(6). Let G be a tree. Since G is connected, between any two distinct vertices of G there exists a path. If there exist more than one path between $u, v \in V(G)$, then by Proposition 9.2.8 any two of these u-v paths will contain a cycle. This is not possible as G is acyclic. Hence the uniqueness of such a path.

Conversely, let (6) hold. Then G is clearly connected. Further, if G has a cycle, then that cycle would provide two paths between any two vertices on the cycle. Hence G is acyclic, *i.e.*, G is a tree. ■

Proposition *The center of a tree is either a singleton or has at most two vertices.*

Proof. Let T be a tree of radius k. Since the center contains at least one vertex, let u be a vertex in the center of T. Now, let v be another vertex in the center. We claim that u is adjacent to v.

On the contrary, suppose u v. Then, there exists a path from u to v, denoted $P(u, v)$, with at least one internal vertex, say w. Let x be any pendant ($d(x) = 1$) vertex of T. Then, either $v \in P(x, w)$ or $v \notin P(x, w)$. In the latter case, check that $\|P(x, w)\| < \|P(x, v)\| \leq k$.

If $v \in P(x, w)$, then $u \notin P(x, w)$ and $\|P(x, w)\| < \|P(x, u)\| \le k$. Thus in either case, the distance from w to any pendant vertex is less than k. Hence, k is not the radius, a contradiction. Thus, $uv \in T$. We cannot have another vertex in the center, or else, we will have a C_3 in T, a contradiction. ■

EXERCISE. *Draw a tree on 8 vertices. Label $V(T)$ as $1, \ldots, 8$ so that each vertex $i \ge 2$ is adjacent to exactly one element of $\{1, 2, \ldots, i-1\}$.*

Proposition. *Let T be a tree on n vertices. Let G be a graph with $\delta(G) \ge n-1$. Then G has a subgraph H with $H \cong T$.*

Proof. We prove the result by induction on n. The result is trivially true if $n = 1$ or 2. So, let the result be true for every tree on $n-1$ vertices and take a tree T on n vertices. Also, suppose that G is any graph with $\delta(G) \ge n-1$.

Due to Corollary 9.4.10, let $v \in V(T)$ with $d(v) = 1$. Take $u \in V(T)$ such that $uv \in E(T)$. Now, consider the tree $T_1 = T - v$. Then, $\delta(G) \ge n-1 > n-2$. Hence, by induction hypothesis, G has a subgraph H such that $H \cong T_1$ under a map, say φ. Let $h \in V(H)$ such that $\varphi(h) = u$. Since $\delta(G) \ge n-1$, h has a neighbor, say h_1, such that h_1 is not a vertex in H but is a vertex in G. Now, map this vertex to v to get the required result. ■

XERCISE. *In the above process, prove that $u_j = i$, for some j, if and only if $d(i) \ge 2$.*

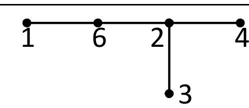
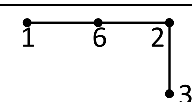
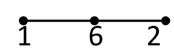
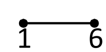
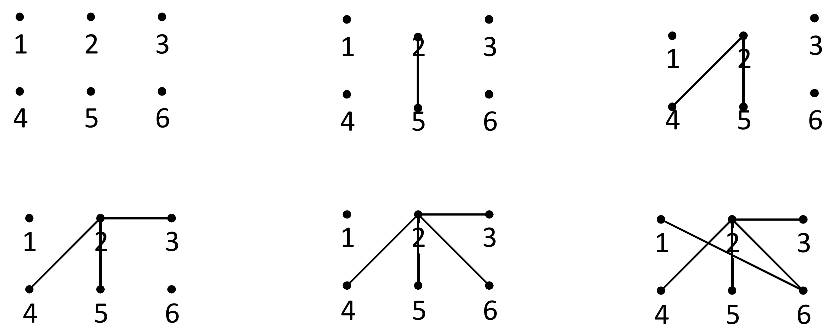
Step	**Pendant v_i**	**Neighbor u_i**	**$P_T = X(1), X(2), \ldots$**	**$T_i = T - v_i$**
1	5	2	2	
2	4	2	2,2	
3	3	2	2,2,2	
4	2	6	2,2,2,6	

Figure : A tree T on 6 vertices

1. At step 1, the vertex 5 was deleted. Hence, $V(T_1) = \{1, 2, 3, 4, 6\}$ with the given sequence 2, 2, 6. So, the pendants in T_1 are $\{1, 3, 4\}$ and the vertex 4 (largest pendant) is adjacent to 2.
2. Now, $V(T_2) = \{1, 2, 3, 6\}$ with the sequence as 2, 6. So, 3 is adjacent to 2.
3. Now, $V(T_3) = \{1, 2, 6\}$ with the sequence as 6. So, the pendants in the current T are $\{1, 2\}$ and 2 is adjacent to 6.
4. Lastly, $V(T_4) = \{1, 6\}$. As the process ends with K_2 and we have only two vertices left, they must be adjacent.

The corresponding set of figures are as follows.



Proposition . *Let T be a tree on the vertex set $\{1, 2, \ldots, n\}$. Then, $d(v) \geq 2$ if and only if v appears in the Prüfer code P_T. Thus, $\{v : v \notin P_T\}$ are precisely the pendant vertices in T.*

Proof. Let $d(v) \geq 2$. Since the process ends with an edge, there is a stage, say i, where $d(v)$ decreases strictly. Thus, at the $(i-1)$-th stage, v was adjacent to a pendant vertex w and at the i-th stage w was deleted and thus, v appears in the sequence.

Conversely, let v appear in the sequence at the k-th stage for the first time. Then, the tree T_k had a pendant vertex w of highest label that was adjacent to v. Note that $T_k - w$ is a tree with at least two vertices. Thus, $d(v) \geq d_{T_k}(v) \geq 2$. ■

EXERCISE *Prove that in the Prüfer code of T a vertex v appears exactly $d(v) - 1$ times. [Hint: Use induction and if v is the largest pendant adjacent to w and $T' = T - v$ then $P_T = w, P_{T'}$.]*

Proposition. *Let T and T' be two trees on the same vertex set of integers. If $P_T = P_{T'}$, then $T = T'$.*

Proof. The statement is trivially true for $|T| = 3$. Assume that the statement holds for $|T| < n$. Now, let T and T' be two trees with vertex set $\{1, 2, \ldots, n\}$ and $P_T = P_{T'}$. As $P_T = P_{T'}$, T and T' have the same set of pendants. Further, the largest labeled pendant w is adjacent to the vertex $X(1)$ in both the trees. Thus, $P_{T-w} = P_{T'-w}$ and hence, by induction hypothesis $T - w = T' - w$. Thus, by PMI, $T = T'$. ■

Proposition *Let S be a set of $n \geq 3$ integers and let X be a sequence of length $n - 2$ of elements from S. Then, there is a tree T with $V(T) = S$ and $P_T = X$.*

Proof. Verify the statement for $|T| = 3$. Now, let the statement hold for all trees T on $n > 3$ vertices and consider a set S of $n + 1$ integers and a sequence X of length $(n-1)$ of elements of S.

Let $v = \max\{x \in S : x \notin X\}$, $S' = S - v$ and $X' = X(2), \ldots, X(n-1)$. By definition, note that $v \neq X(i)$, for $2 \leq i \leq n - 1$. Thus, X' is a sequence of elements of S' of length $n - 2$. As $|S'| = n$, by induction hypothesis, there is a tree T' with $P_{T'} = X'$.

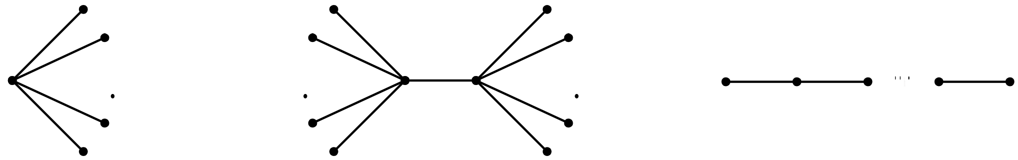
Let T be the tree obtained by adding a new pendant v at the vertex $X(1)$ of T'. In T', the vertices $X(i)$, for $i \geq 2$, were not available as pendants and now in T the vertex $X(1)$ is also not available as a pendant (here some $X(i)$'s may be the same). Let $R' = \{x \in S' : x \notin X'\}$ be the pendants in T'. Then, the set of pendants in T is $(R' \cup \{v\}) \setminus \{X(1)\}$ which equals $\{x \in S : x \notin X\}$. Thus, v is the pendant of T of maximum label. Hence, $P_T = X$. ■

Theorem. [A. Cayley, 1889, Quart. J. Math] *Let $n \geq 3$. Then, there are n^{n-2} different trees with vertex set $\{1, 2, \ldots, n\}$.*

Proof. Let F be the class of trees on the vertex set $\{1, 2, \ldots, n\}$ and let G be the class of $(n-2)$-sequences of $\{1, 2, \ldots, n\}$. Note that the function $f : F \longrightarrow G$ defined by $f(T) = P_T$, the Prüfer code, is a one-one and onto mapping. As $|G| = n^{n-2}$, the required result follows. ■

EXERCISE **9.4.24.** 1. *Find out all non-isomorphic trees of order 6 or less.*

2. *Count with diameter: how many non-isomorphic trees are there of order 7?*
3. *Show that every automorphism of a tree fixes a vertex or an edge.*
4. *Give a class of trees T with $|\Gamma(T)| = 6$.*
5. *Let T be a tree, $\sigma \in \Gamma(T)$, $u \in V(T)$ such that $\sigma^2(u) \neq u$. Can we have an edge $uv \in E(T)$ such that $\sigma(u) = v$?*
6. *Let T be a tree with center $\{u\}$ and radius r. Let v satisfy $d(u, v) = r$. Show that $d(v) = 1$.*
7. *Let T be a tree with $|T| > 2$. Let T' be obtained from T by deleting all the pendant vertices of T. Show that the center of T is the same as the center of T'.*
8. *Let T be a tree with center $\{u\}$ and $\sigma \in \Gamma(T)$. Show that $\sigma(u) = u$.*
9. *Is it possible to have a tree such that $|\Gamma(T)| = 7$?*
10. *Construct a tree T on vertices $S = \{1, 2, 3, 6, 7, 8, 9\}$ for which $P_T = 6, 3, 7, 1, 2$.*
11. *Draw the tree on the vertex set $\{1, 2, \ldots, 12\}$ whose Prüfer code is 9954449795.*
12. *Practice with examples: get the Prüfer code from a tree; get the tree from a given code and a vertex set.*



13. *How many trees of the following forms are there on the vertex set* $\{1, 2, . . . , 100\}$*?*
14. *Show that any tree has at least* $\Delta(T)$ *leaves (pendant edges).*
15. *Let* T *be a tree and* T_1, T_2, T_3 *be subtrees of* T *such that* $T_1 \cap T_3 \neq \emptyset$, $T_2 \cap T_3 \neq \emptyset$ *and* $T_1 \cap T_2 \cap T_3 = \emptyset$. *Show that* $T_1 \cap T_2 = \emptyset$.
16. *Let* $\mathcal{F}$ *be a set of subtrees of a tree* T. *Assume that the trees in* $\mathcal{F}$ *have nonempty pairwise intersection. Show that their overall intersection is nonempty. Is this true, if we replace* T *by a graph* G*?*
17. *A connected graph* G *is said to be* **unicyclic** *if* G *has exactly one cycle as its subgraph. Prove that if* G *is connected and* $|G| = \|G\|$, *then* G *is a unicyclic graph.*

Eulerian graphs

Definition 9.5.1. Let G be a graph. Then, G is said to have an **Eulerian tour** if there is a closed walk, say $[v_0, v_1, . . . , v_k, v_0]$, such that each edge of the graph appears exactly once in the walk. The graph G is said to be **Eulerian** if it has an Eulerian tour.

Note that by definition, a disconnected graph is not Eulerian. In this section, the graphs can have loops and multiple edges. The graphs that have a closed walk traversing each edge exactly once have been named “Eulerian graphs” due to the solution of the famous Königsberg bridge problem by Euler in 1736. The problem is as follows: The city of Königsberg (the present day Kaliningrad) is divided into 4 land masses by the river Pregolya. These land masses are joined by 7 bridges (see Figure 9.11). The question required one to answer “is there a way to start from a land mass that passes through all the seven bridges in Figure 9.11 and return back to the starting land mass”? Euler, rephrased the problem along the following lines: Let the four land masses be denoted by the vertices A, B, C and D of a graph and let the 7 bridges correspond to 7 edges of the graph. Then, he asked “does this graph have a closed walk that traverses each edge exactly once”? He gave a necessary and sufficient condition for a graph to have such a closed walk and thus giving a negative answer to Königsberg bridge problem.

One can also relate the above problem to the problem of “starting from a certain point, draw a given figure with pencil such that neither the pencil is lifted from the paper nor a line is repeated such that the drawing ends at the initial point”.

Theorem . [Euler, 1736] *A connected graph is Eulerian if and only if each vertex in the graph is of even degree.*

Proof. Let G be a connected graph. Suppose G has an Eulerian tour, say $W = [v_0, v_1, . . . , v_k, v_0]$. Observe that whenever we arrive at a vertex $v \neq v_0$ using an edge, say e, in W then we leave that vertex using an edge, say e' in W with $e \neq e'$. As each edge appears exactly once in W and each edge is traversed, $d(v) = 2r$, if v appears r times in the tour. Also, if v_0 appears r times in the tour then $d(v_0) = 2(r - 1)$. Hence, $d(v)$ is always even.

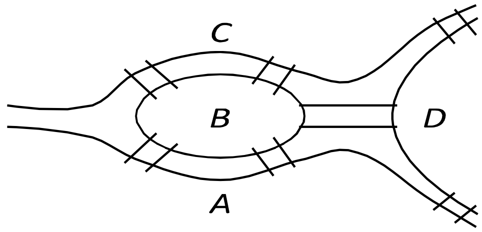


Figure 9.11: Königsberg bridge problem

Conversely, let G be a connected graph with each vertex having even degree. Let $W = v_0v_1 \cdots v_k$ be a longest walk in G without repeating any edge in it. As v_k has an even degree it follows that $v_k = v_0$, otherwise W can be extended. If W is not an Eulerian tour then there exists an edge, say $e' = v_iw$, with $w \neq v_{i-1}, v_{i+1}$. In this case, $W' = wv_i \cdots v_k(= v_0)v_1 \cdots v_{i-1}v_i$ is a longer tour compared to W, a contradiction. Thus, there is no edge lying outside W and hence W is an Eulerian tour. ■

Proposition *Let G be a connected graph with exactly two vertices of odd degree. Then, there is an Eulerian walk starting at one of those vertices and ending at the other.*

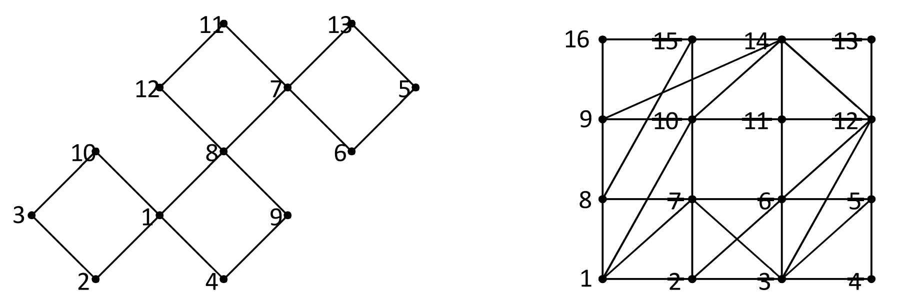
Proof. Let x and y be the two vertices of odd degree and let v be a symbol such that $v \notin V(G)$. Then, the graph H with $V(H) = V(G) \cup \{v\}$ and $E(H) = E(G) \cup \{xv, yv\}$ has each vertex of even degree and hence by Theorem 9.5.2, H is Eulerian. Let $\Gamma = [v, v_1 = x, \ldots, v_k = y, v]$ be an Eulerian tour. Then, $\Gamma - v$ is an Eulerian walk with the required properties. ■

EXERCISE **.** *Let G be an Eulerian graph and let e be any edge. Show that $G - e$ is connected.*

How to find an Eulerian tour (algorithm)?

Start from a vertex v_0, move via edge that has not been taken and go on deleting them. <u>Do not take an edge</u> whose deletion creates a non-trivial component not containing v_0.

EXERCISE **9.5.5.** *Find Eulerian tours for the following graphs.*



Theorem . [Finding Eulerian tour] *The previous algorithm correctly gives an Eulerian tour provided the given graph is Eulerian.*

Proof. Let the algorithm start at a vertex, say v_0. Now, assume that we are at u with H as the current graph and C as the only non-trivial component of H. Thus, $d_H(u) > 0$. Assume that the deletion of the edge uv creates a non-trivial component not containing v_0. Let C_u and C_v be the components of $C - uv$, containing u and v, respectively.

We first claim that $u \neq v_0$. In fact, if $u = v_0$, then H must have all vertices of even degree and $d_H(v_0) \geq 2$. So, C is Eulerian. Hence, $C - uv$ cannot be disconnected, a contradiction to $C - uv$ having two components C_u and C_v. Thus, $u \neq v_0$. Moreover, note that the only vertices of odd degree in C is u and v_0.

Now, we claim that C_u is a non-trivial component. Suppose C_u is trivial. Then, $v_0 \in C_v$, a contradiction to the assumption that the deletion of the edge uv creates a non-trivial component not containing v_0. So, C_u is non-trivial.

Finally, we claim that $v_0 \in C_v$. If possible, let $v_0 \in C_u$. Then, the only vertices in $C - uv$ of odd degree are $v \in C_v$ and $v_0 \in C_u$. Hence, $C - uv + v_0v$ is a connected graph with each vertex of even degree. So, by Theorem 9.5.2, the graph $C - uv + v_0v$ is Eulerian. But, this cannot be true as vv_0 is a bridge. Thus, $v_0 \in C_v$.

Hence, C_u is the newly created non-trivial component not containing v_0. Also, each vertex of C_u has even degree and hence by Theorem 9.5.2, C_u is Eulerian. This means, we can take an edge e' incident on u and complete an Eulerian tour in C_u. So, at u if we take the edge e' in place of the edge e, then we will not create a non-trivial component not containing v_0.

Thus, at each stage of the algorithm either $u = v_0$ or there is a path from u to v_0. Moreover, this is the only non-trivial connected component. When the algorithm ends, we must have $u = v_0$. Because, as seen above, the condition $u \neq v_0$ gives the existence of an edge that is incident on u and that can be traversed (as $d_H(u)$ is odd). Hence, if $u \neq v_0$, the algorithm cannot stop. Thus, when algorithm stops $u = v_0$ ■ and all components are trivial.

EXERCISE **9.5.7.**

1. *Apply the algorithm to graphs of Exercise 9.5.5. Also, create connected graphs, where each vertex is of even degree, and apply the above algorithm.*

2. *Give a necessary and sufficient condition on m and n so that $K_{m,n}$ is Eulerian.*
3. *Each of the 8 persons in a room has to shake hands with every other person as per the following rules:*
   - *(a) The handshakes should take place sequentially.*
   - *(b) Each handshake (except the first) should involve someone from the previous handshake.*
   - *(c) No person should be involved in 3 consecutive handshakes.*

   Is there a way to sequence the handshakes so that these conditions are all met?
4. *Prove: A connected graph G is Eulerian if and only if the $E(G)$ can be partitioned into cycles.*

Hamiltonian graphs

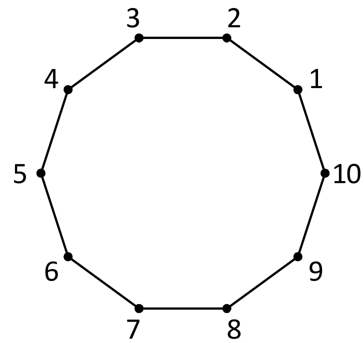
Definition . let G be a graph. A cycle in G is said to be **Hamiltonian** if it contains all vertices of G. If G has a Hamiltonian cycle, then G is called a **Hamiltonian** graph.

Finding a nice characterization of a Hamiltonian graph is an unsolved problem.

Example

1. For each positive integer $n \geq 3$, the cycle C_n is Hamiltonian.
2. The graphs corresponding to all platonic solids are Hamiltonian.
3. The Petersen graph is a non-Hamiltonian Graph (the proof appears below).

Proposition . *The Petersen graph is not Hamiltonian.*



A Hamiltonian graph

Proof. Suppose that the Petersen graph, say G, is Hamiltonian. So, G contains $C_{10} = [1, 2, 3, \ldots, 10, 1]$ as a subgraph. As each vertex of G has degree 3, $G = C_{10} + M$, where M is a set of 5 chords in which each vertex appears as an endpoint. Now, consider the vertices 1, 2 and 3.

Since, $g(G) = 5$, the vertex 1 can be adjacent to only one of the vertices 5, 6 or 7. Hence, if 1 is adjacent to 5, then the possible third vertex that is adjacent to 10 will create cycles of length 3 or 4. Similarly, if 1 is adjacent to 7 then there is no choice for the possible third vertex that can be adjacent to 2. So, let 1 be adjacent to 6. Then, 2 must be adjacent to 8. In this case, note that there is no choice for the third vertex that can be adjacent to 3. Thus, the Petersen graph is non-Hamiltonian. ■

Theorem . *Let G be a Hamiltonian graph. Then, for $S \subseteq V(G)$ with $S \neq \emptyset$, the graph $G - S$ has at most $|S|$ components.*

Proof. Note that by removing k vertices from a cycle, one can create at most k connected components. Hence, the required result follows. ■

Theorem [Dirac, 1952] *Let G be a graph with $|G| = n \geq 3$ and $d(v) \geq n/2$, for each $v \in V(G)$. Then G is Hamiltonian.*

Proof. We first show that G is connected. If possible, let G be disconnected. Then G has a component, say H, with $|V(H)| = k \leq n/2$. Hence, $d(v) \leq k - 1 < n/2$, for each $v \in V(H)$. A contradiction to $d(v) \geq n/2$, for each $v \in V(G)$. Therefore, G is connected.

Now, let $P = [v_1, v_2, \cdots, v_k]$ be a longest path in G. Since P is a longest path, all neighbors of v_1 and v_k are in P and $k \leq n$. We claim that there exists an i such that $v_1 \sim v_i$ and $v_{i-1} \sim v_k$. Otherwise, for each $v_i \sim v_1$, we must have v_{i-1} v_k. Then, $|N(v_k)| \leq k - 1 - |N(v_1)|$. Hence, $|N(v_1)| + |N(v_k)| \leq k - 1 < n$, a contradiction to $d(v) \geq n/2$ for each $v \in V(G)$.

So, the claim is valid and hence, we have a cycle $\tilde{P} := v_1 v_i v_{i+1} \cdots v_k v_{i-1} \cdots v_1$ of length k.

We now prove that $\tilde{P}$ gives a Hamiltonian cycle. Suppose not. Then, there exists $v \in V(G)$ such that v is outside P and v is adjacent to some v_j. Now, use $\tilde{P}$, v and v_j to create a path whose length is larger than the length of P. Hence, P cannot be a path of longest length, a contradiction. Thus, the required result follows. ■

A slight relaxation on the sufficient condition of a graph to be Hamiltonian is provided by the following result. We expect the reader to prove it.

Theorem . [Ore, 1960] *Let G be a graph on $n \geq 3$ vertices such that $d(u) + d(v) \geq n$ for every pair of non-adjacent vertices u and v. Then G is Hamiltonian.*

Lemma . *Let u and v be two non-adjacent vertices of a graph G such that $d(u) + d(v) \geq |G|$. Then G is Hamiltonian if and only if $G + uv$ is Hamiltonian.*

Proof. If G is Hamiltonian, then so is $G+uv$. Conversely, suppose that $G+uv$ is Hamiltonian. If $G+uv$ has a Hamiltonian cycle not using uv, then G is Hamiltonian. Otherwise, let $[u = v_1, \ldots v_n = v, u]$ be a Hamiltonian cycle in $G + uv$. Then, the path $[v_1, \ldots, v_n]$ is available in G. Now proceeding as in the proof of Dirac's theorem, as $d(v_1) + d(v_n) \geq n$, we see that there must exist an i such that $v_1 \sim v_i$ and $v_n \sim v_{i-1}$. Then the cycle $[v_1, v_i, v_{i+1}, \ldots, v_n, v_{i-1}, v_{i-2}, \ldots, v_1]$ is a Hamiltonian cycle in G. ■

Discussion. **[Closure]** Let G be a graph on n vertices, $n \geq 2$. Suppose we perform the following operation(s) on G.

Step 1: If G has two nonadjacent vertices $u \neq v$ such that $d(u) + d(v) \geq n$, then add the edge (u, v) in G and treating the resulting graph as G, repeat Step 1, until the graph has no nonadjacent vertices $u \neq v$ satisfying $d(u) + d(v) \geq n$.

Step 2: If G has no nonadjacent vertices $u \neq v$ such that $d(u) + d(v) \geq n$, then stop.

For example, let G be the trivial graph on 10 vertices (G has no edge). Then, the application of the above operation stops with the trivial graph itself. Whereas, if G is the graph obtained from K_{10} by deleting the edges $\{1, 2\}$ and $\{3, 4\}$, then applying the above operation gives K_{10} as the result.

Notice that in the above example, one might have added the edge $\{1, 2\}$ first and then the edge $\{3, 4\}$ whereas some one else might have added $\{3, 4\}$ first and then $\{1, 2\}$. However, they both get the same end result. We prove this for any graph G. Before that, note that, if G is any graph on n vertices, then the above operation can add at most a finitely many edges as the end result has to be a subgraph of K_n.

Proposition *Let G be a graph on n vertices. Suppose the application of the operations described in Discussion 9.6.8 to G by following two different sequences of edge additions gives K and F as the end results. Then $K = F$.*

Proof. Let K and F be obtained by sequentially adding edges

$$(e\text{-list}) \qquad e_1 = u_1v_1, \ldots, e_k = u_kv_k \qquad \text{and} \qquad (f\text{-list}) \qquad f_1 = x_1y_1, \ldots, f_r = x_ry_r,$$

respectively, to G in that order.

Assume that $K \neq F$. Then, without loss of generality, suppose an edge has been added in the e-list which doesn't appear in the f-list. Let e_i be the first such edge in the e-list which does not appear in the f-list. Put $H = G + e_1 + \cdots + e_{i-1}$. As $e_1, \ldots, e_{i-1}$ are in the f-list, we see that H is a subgraph of F.

Furthermore, taking $e_i = \{u, v\}$, as e_i was the next to be added in the e-list, it follows that $d_H(u) + d_H(v) \geq n$. But as H is a subgraph of F, we see that $d_F(u) + d_F(v) \geq n$ too.

This means that F is not the end result, because in an end result there are no nonadjacent vertices $u \neq v$ with sum of degrees at least n. This is a contradiction. ■

Let G be a graph. The graph obtained as the end result of applying the operation described in Discussion 9.6.8, is called the **closure of** G, denoted $C(G)$. (It is obtained by repeatedly choosing pairs of non-adjacent vertices u, v such that $d(u) + d(v) \geq |G|$ and adding edges between them until no such pair of vertices exist.) Proposition 9.6.9 tells us that for any graph G, $C(G)$ is unique.

Corollary. *Let G be a graph. Then G is Hamiltonian if and only if $C(G)$ is Hamiltonian. In particular, if $C(G)$ is Hamiltonian, then G is Hamiltonian.*

Proof. Follows from Lemma 9.6.7. ■

QUIZ. *Let G be a graph on $n \geq 3$ vertices. If G has a cut-vertex, then prove that $C(G) \neq K_n$.*

Theorem. *Let $d_1 \le \cdots \le d_n$ be the vertex degrees of G which satisfy the property*

R : If $d_k \le k$ then $d_{n-k} \ge n - k$ for each $k < n/2$.

Then G is Hamiltonian.

Proof. We show that under the above condition $H = C(G) \cong K_n$. On the contrary, assume that there exists a pair of vertices $u, v \in V(G)$ such that $uv \notin E(H)$ and $d_H(u) + d_H(v) \le n - 1$. Among all such pairs, choose a pair $u, v \in V(G)$ such that $uv \notin E(H)$ and $d_H(u) + d_H(v)$ is maximum. Assume that $d_H(v) \ge d_H(u) = k$ (say). As $d_H(u) + d_H(v) \le n - 1$, we get $k < n/2$.

Now, let $S_v = \{x \in V(H) : x \neq v, xv \notin E(H)\}$ and $S_u = \{w \in V(H) : w \neq u, wu \notin E(H)\}$. Therefore, the assumption that $d_H(u) + d_H(v)$ is the maximum among each pair of vertices u, v with $uv \notin E(H)$ and $d_H(u) + d_H(v) \le n - 1$ implies that $|S_v| = n - 1 - d_H(v) \ge d_H(u) = k$ and $d_H(x) \le d_H(u) = k$, for each $x \in S_v$. So, there are at least k vertices in H (elements of S_v) with degrees at most k.

Also, for any $w \in S_u$, note that the choice of the pair u, v implies that $d_H(w) \le d_H(v) \le n - 1 - d_H(u) = n - 1 - k < n - k$. As $d_H(u) = k$, $|S_u| = n - 1 - k$. Further, the condition $d_H(u) + d_H(v) \le n - 1$, $d_H(v) \ge d_H(u) = k$ and $u \notin S_u$ implies that $d_H(u) \le n - 1 - d_H(v) \le n - 1 - k < n - k$. So, there are $n - k$ vertices ($S_u \cup \{u\}$) in H with degrees less than $n - k$.

Therefore, if $d'_1 \le \cdots \le d'_n$ are the vertex degrees of H, then we observe that there exists a $k < n/2$ for which $d'_k \le k$ and $d'_{n-k} < n - k$. As $k < n/2$ and $d_i \le d'_i$, we get a contradiction to the given hypothesis. ■

EXERCISE *Let $d_1 \le \cdots \le d_n$ be the vertex degrees of G which satisfy the property R (see Theorem). Then show that $C(G)$ also has property R.*

Definition. The **line graph** H of a graph G is a graph with $V(H) = E(G)$ and $e_1, e_2 \in V(H)$ are adjacent in H if e_1 and e_2 share a common vertex/endpoint.

Example Verify the following:

1. Line graph of C_5 is C_5.
2. Line graph of P_5 is P_4.
3. Line graph of any graph G contains a complete subgraph of size $\Delta(G)$.

EXERCISE *1. Let G be a connected Eulerian graph. Show that the line graph of G is Hamiltonian. Is the converse true?*

2. What can you say about the clique number of a line graph?

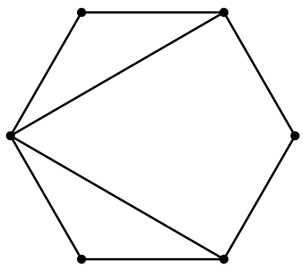
Theorem *A connected graph G is isomorphic to its line graph if and only if $G = C_n$ for some $n \ge 3$.*

Proof. If G is isomorphic to its line graph, then $|G| = \|G\|$. Thus, G is a unicyclic graph. Let $[v_1, v_2, \ldots, v_k, v_{k+1} = v_1]$ form the cycle in G. Then, the line graph of G contains a cycle $P = [v_1v_2, v_2v_3, \ldots, v_kv_1]$. We now claim that $d_G(v_i) = 2$.

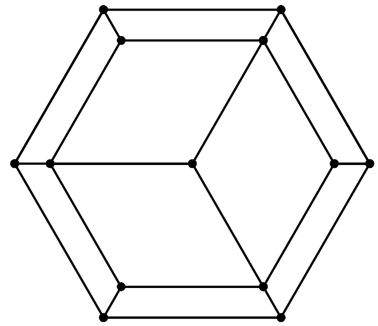
Suppose not and let $d_G(v_1) \ge 3$. So, there exists a vertex $u \notin \{v_2, \ldots, v_k\}$ such that $u \sim v_1$. In that case, the line graph of G contains the triangle $T = [v_1v_2, v_1v_k, v_1u]$ and $P \neq T$. Thus, the line graph is not unicyclic, a contradiction. ■

EXERCISE *1. Consider the graphs shown below.*

(a) Determine the closure of G.



G



H

(b) *Show that H is not Hamiltonian.*

2. *Give a necessary and sufficient condition on $m, n \in \mathbf{N}$ so that $K_{m,n}$ is Hamiltonian.*
3. *Show that any graph with at least 3 vertices and atleast $\binom{n-1}{2} + 2$ edges is Hamiltonian.*
4. *Show that for any $n \geq 3$ there is a graph H with $\|G\| = \binom{n-1}{2} + 1$ that is not Hamiltonian. But, prove that all such graphs H admit a Hamiltonian path (a path containing all vertices of H).*

Bipartite graphs

Definition 9.7.1. A graph is said to be 2-**colorable** if its vertices can be colored with two colors in a way that adjacent vertices get different colors.

Example 9.7.2. Prove the following results.

16. Every tree is 2-colorable.
17. Every cycle of even length is 2-colorable.
18. The complete bipartite graphs, namely $K_{m,n}$, are 2-colorable
19. Petersen graph is not 2-colorable but 3-colorable.

Lemma. *Let P and Q be two v-w paths in G such that length of P is odd and length of Q is even. Then, G contains an odd cycle.*

Proof. If P, Q have no inner vertex (a vertex other than v, w) in common then $P \cup Q$ is an odd cycle in G.

So, suppose P, Q have an inner vertex in common. Let x be the first common inner vertex when we walk from v to w. Then, one of $P(v, x)$, $P(x, w)$ has odd length and the other is even. Let $P(v, x)$ be odd. If length of $Q(v, x)$ is even then $P(v, x) \cup P(x, v)$ is an odd cycle in G. If length of $Q(v, x)$ is odd then the length of $Q(x, w)$ is also odd and hence we can consider the x-w paths $P(x, w)$ and $Q(x, w)$ and proceed as above to get the required result. ∎

Theorem . *Let G be a connected graph with at least two vertices. Then the following statements are equivalent:*

1. *G is 2-colorable.*
2. *G is bipartite.*
3. *G does not have an odd cycle.*

Proof. (1)⇒ (2). Let G be 2-colorable. Let V_1 be the set of red vertices and V_2 be the set of blue vertices. Clearly, G is bipartite with partition V_1, V_2.

(2)⇒ (1). Color the vertices in V_1 with red color and that of V_2 with blue color to get the required 2 colorability of G.

(2) ⇒ (3). Let G be bipartite with partition V_1, V_2. Let $v_0 \in V_1$ and suppose $\Gamma = v_0v_1v_2\cdots v_k = v_0$ is a cycle. It follows that $v_1, v_3, v_5 \cdots \in V_2$. Since $v_k \in V_1$, we see that k is even. Thus, Γ has an even length.

(3) ⇒ (2). Suppose that G does not have an odd cycle. Pick any vertex v. Define

$$V_1 = \{w : \text{there is a walk of even length from } v \text{ to } w\}$$
$$V_2 = \{w : \text{there is a walk of odd length from } v \text{ to } w\}.$$

Clearly, $v \in V_1$. Also, G does not have an odd cycle implies that $V_1 \cap V_2 = \emptyset$ (use Lemma 9.7.3). As G is connected each w is either in V_1 or in V_2.

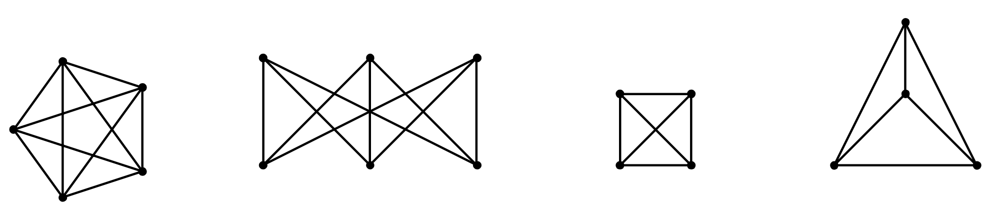
Let $x \in V_1$. Then, there is an even path $P(v, x)$ from v to x. If $xy \in E(G)$, then we have a v-y walk of odd length. Deleting all cycles from this walk, we have an odd v-y path. Thus, $y \in V_2$. Similarly, if $x \in V_2$ and $xy \in E$, then $y \in V_1$. Thus, G is bipartite with parts V_1, V_2. ■

EXERCISE. 1. *There are* 15 *women and some men in a room. Each man shook hands with exactly* 6 *women and each woman shook hands with exactly* 8 *men. How many men are there in the room?*

2. *How do you test whether a graph is bipartite or not?*
3. *Prove the statements in Example 9.7.2.*
4. *Let G and H be two bipartite graphs. Prove that G × H is also a bipartite graph.*

Planar graphs

Definition. A graph is said to be **embedded** on a surface S when it is drawn on S so that no two edges intersect. A **plane graph** is a graph drawn on the plane where no two edges intersect. A graph is said to be **planar** if it can be embedded on the plane.



K_5-Non-planar $K_{3,3}$-Non-planar K_4 K_4 - Planar embedding

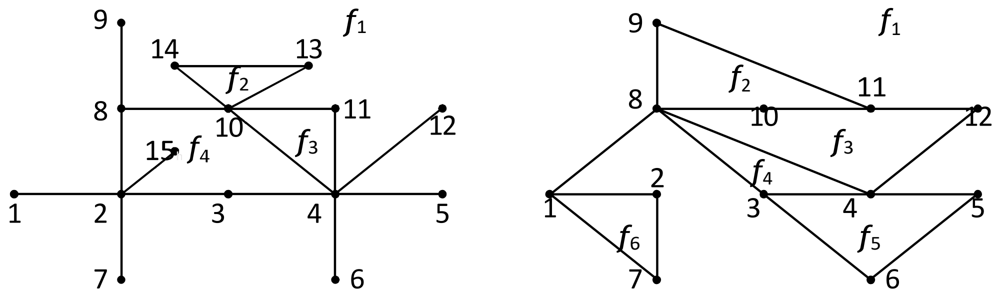
Figure 9.13: Planar and non-planar graphs

1. A tree is embeddable on a plane..
2. Any cycle C_n, $n \geq 3$ is planar.
3. The planar embedding of K_4 is given in Figure 9.13.
4. Draw a planar embedding of $K_{2,3}$.
5. Draw a planar embedding of the edges of a three dimensional cube.

6. Draw a planar embedding of $K_5 - e$, where e is any edge.
7. Draw a planar embedding of $K_{3,3} - e$, where e is any edge.

Definition onsider a planar embedding of a graph G. The regions on the plane defined by this embedding are called **faces/regions** of G. The unbounded face/region is called the exterior face (see Figure 9.14).

Example. Consider the following planar embedding of the graphs X_1 and X_2.



Planar Graph X_1 Planar Graph X_2

Planar graphs with labeled faces to understand the Euler's theorem

1. The faces of the planar graph X_1 and their corresponding edges are listed below.

Face	Corresponding Edges
f_1	$\{9, 8\}$, $\{8, 9\}$, $\{8, 2\}$, $\{2, 1\}$, $\{1, 2\}$, $\{2, 7\}$, $\{7, 2\}$, $\{2, 3\}$, $\{3, 4\}$, $\{4, 6\}$, $\{6, 4\}$, $\{4, 5\}$, $\{5, 4\}$, $\{4, 12\}$, $\{12, 4\}$, $\{4, 11\}$, $\{11, 10\}$, $\{10, 13\}$, $\{13, 14\}$, $\{14, 10\}$, $\{10, 8\}$, $\{8, 9\}$
f_2	$\{10, 13\}$, $\{13, 14\}$, $\{14, 10\}$
f_3	$\{4, 11\}$, $\{11, 10\}$, $\{10, 4\}$
f_4	$\{2, 3\}$, $\{3, 4\}$, $\{4, 10\}$, $\{10, 8\}$, $\{8, 2\}$, $\{2, 15\}$, $\{15, 2\}$

2. Determine the faces of the planar graph X_2 and their corresponding edges.
3. Any planar embedding of a tree has only one face, the exterior face.
4. Any planar embedding of a cycle has two faces.

From the table, we observe that each edge of X_1 appears in two faces. This is easily seen for the faces that do not have pendant vertices (see the faces f_2 and f_3). In faces f_1 and f_4, there are a few edges which are incident with a pendant vertex. Notice that the edges that are incident with a pendant vertex, *e.g.*, the edges $\{2, 15\}$, $\{8, 9\}$ and $\{1, 2\}$ etc., appear twice when traversing a particular face. This observation leads to the proof of Euler's theorem for planar graphs which is stated next.

Theorem [Euler Formula] *Let G be a connected plane graph with f number of faces. Then*

$$|G| - \|G\| + f = 2. \tag{9.1}$$

Proof. We use induction on f. Let $f = 1$. Then G cannot have a subgraph isomorphic to a cycle. For, if G has a subgraph isomorphic to a cycle, then in any planar embedding of G, $f \geq 2$. Therefore, G is a tree; and hence $|G| - \|G\| + f = n - (n-1) + 1 = 2$.

Assume that Equation (9.1) is true for all plane connected graphs having $2 \leq f < n$. Let G be a connected planar graph with $f = n$. Choose an edge that is not a cut-edge, say e. Then, $G - e$ is st a connected graph. Notice that the edge e is incident with two separate faces. So, its removal will combine the two faces, and hence $G-e$ has only $n-1$ faces. Thus, using the induction hypothesis

$$|G|-\|G\|+f=|G-e|-(\|G-e\|+1)+n=|G-e|-\|G-e\|+(n-1)=2.$$

Hence the required result follows. ■

Lemma *Let G be a plane bridgeless graph with $\|G\| \geq 2$. Then $2\|G\| \geq 3f$. Further, if G has no cycle of length 3, then $2\|G\| \geq 4f \Leftrightarrow \|G\| \geq 2f$.*

Proof. For each edge put two dots on either side of the edge. The total number of dots is $2\|G\|$. If G has a cycle then each face has at least three edges. So, the total number of dots is at least $3f$. Further, if G does not have a cycle of length 3, then $2\|G\| \geq 4f$. ■

Theorem *The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are not planar.*

Proof. If K_5 is planar, then consider a plane representation of it. By Equation (9.1), $f=7$. But, by Lemma 9.8.6, one has $20=2\|G\| \geq 3f=21$, a contradiction.

If $K_{3,3}$ is planar, then consider a plane representation of it. Note that it does not have a C_3. Also, by Euler's formula, $f=5$. Hence, by Lemma 9.8.6, one has $18=2\|G\| \geq 4f=20$, a contradiction. ■

Definition Let G be a graph. Then, a **subdivision** of an edge uv in G is obtained by replacing the edge by two edges uw and wv, where w is a new vertex. Two graphs are said to be **homeomorphic** if they can be obtained from the same graph by a sequence of subdivisions.

For example, the paths P_n and P_m are homeomorphic for all $m, n \in$ **N**. Similarly, all the cyclic graphs are homeomorphic to the cycle C_3. (We are considering only simple graphs. In general, one can say that all cyclic graphs are homeomorphic to the graph $G=(V,E)$, where $V=\{v\}$ and $E=\{\{v,v\}\}$. It is a graph having exactly one vertex and a loop). Also, note that if two graphs are isomorphic then they are also homeomorphic. Figure 9.15 gives examples of homeomorphic graphs that are different from a path or a cycle.

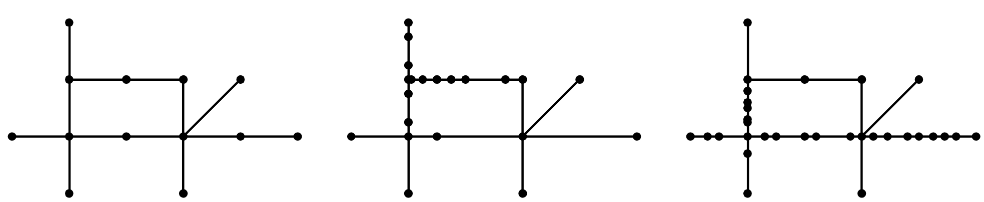


Figure 9.15: Homeomorphic graphs

The following result characterizes planar graphs via homeomorphisms, which we do not prove.

Theorem **[Kuratowski,** 1930**]** *A graph is planar if and only if it has no subgraph homeomorphic to K_5 or $K_{3,3}$.*

We have the following observations that directly follow from Kuratowski's theorem.

Remark 1. Among all simple connected non-planar graphs

(a) the complete graph K_5 has minimum number of vertices.

(b) the complete bipartite graph $K_{3,3}$ has minimum number of edges.

2. If Y is a non-planar subgraph of a graph X then X is also non-planar.

Definition Let G be a graph. Define a relation on the edges of G by $e_1 \sim e_2$ if either $e_1 = e_2$ or there is a cycle containing both these edges. Note that this is an equivalence relation. Let E_i be the equivalence class containing the edge e_i. Also, let V_i denote the endpoints of the edges in E_i. Then, the induced subgraphs $\langle V_i \rangle$ are called the **blocks** of G.

The following result, which we do not prove, characterizes planar graphs via blocks.

Proposition *A graph G is planar if and only if each of its blocks are planar.*

Definition. A graph is called **maximal planar** if it is planar and addition of any more edges results in a non-planar graph.

Notice that a maximal planar graph is necessarily connected.

Proposition. *If G is a maximal planar graph with at least 3 vertices, then every face is a triangle and $\|G\| = 3|G| - 6$.*

Proof. Suppose there is a face, say f, described by the cycle $[u_1, \ldots, u_k, u_1]$, $k \geq 4$. Then, we can take a curve joining the vertices u_1 and u_3 lying totally inside the region f, so that $G + u_1u_3$ is planar. This contradicts the fact that G is maximal planar. Thus, each face is a triangle. It follows that $2\|G\| = 3f$. As $|G| - \|G\| + f = 2$, we have $2\|G\| = 3f = 3(2 - |G| + \|G\|)$ or $\|G\| = 3|G| - 6$.

■

Exercise. 1. *Prove/disprove: A two colorable graph is necessarily planar.*

2. *Suppose that G is a plane graph such that each face is a 4-cycle. What is the number of edges in G?*
3. *Show that the Petersen graph has a subgraph homeomorphic to $K_{3,3}$.*
4. *Show that a plane graph with at least 3 vertices can have at most $2|G| - 5$ bounded faces.*
5. *Let G be a plane graph with f faces and k components. Prove that $|G| - \|G\| + f = k + 1$ (use induction).*
6. *If G is a plane graph without 3-cycles, then show that $\delta(G) \leq 3$.*
7. *Is it necessary that a plane graph G should contain a vertex of degree less than 5?*
8. *Show that any plane graph with at least 4 vertices has a vertex of degree at most five.*
9. *Show that any plane graph with at least 4 vertices has at least four vertices of degree at most 5.*
10. *Produce a planar embedding of the graph G given in Figure 9.16.*

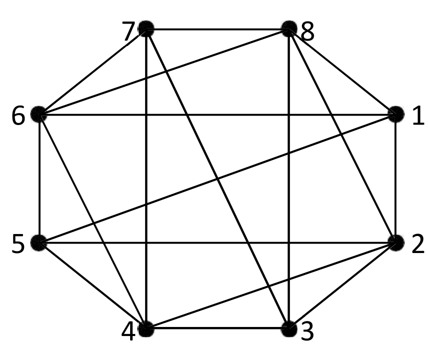


Figure: A graph on 8 vertices

Applications of BFS algorithm

The applications of breadth-first-algorithm are given as follows -

- BFS can be used to find the neighboring locations from a given source locatio n.
- In a peer-to-peer network, BFS algorithm can be used as a traversal method to find all the neighboring nodes. Most torrent clients, such as BitTorrent, uTorrent, etc. emp loy this process to find "seeds" and "peers" in the network.
- BFS can be used in web crawlers to create web page indexes. It is one of the m ain algorithms that can be used to index web pages. It starts traversing from th e source page and follows the links associated with the page. Here, every web page is considered as a node in the graph.
- BFS is used to determine the shortest path and minimum spanning tree.
- BFS is also used in Cheney's technique to duplicate the garbage collection.
- It can be used in ford-Fulkerson method to compute the maximum flow in a flow network.

Algorithm

The steps involved in the BFS algorithm to explore a graph are given as follows -

Step 1: SET STATUS = 1 (ready state) for each node in G

Step 2: Enqueue the starting node A and set its STATUS = 2 (waiting state)

Step 3: Repeat Steps 4 and 5 until QUEUE is empty

Step 4: Dequeue a node N. Process it and set its STATUS = 3 (processed state).

Step 5: Enqueue all the neighbours of N that are in the ready state (whose STA TUS = 1) and set

their STATUS = 2

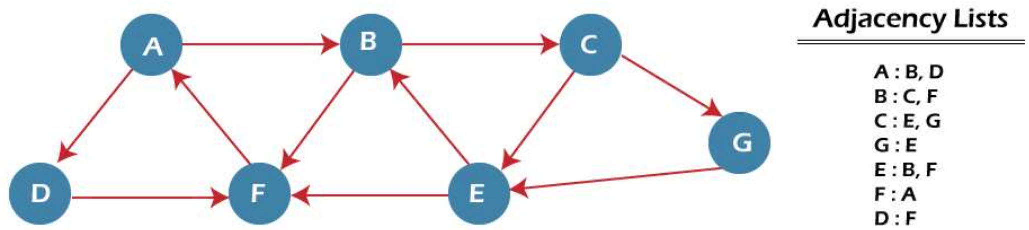
(waiting state)

[END OF LOOP]

Step 6: EXIT

Example of BFS algorithm

Now, let's understand the working of BFS algorithm by using an example. In the example given below, there is a directed graph having 7 vertices.



In the above graph, minimum path 'P' can be found by using the BFS that will start from Node A and end at Node E. The algorithm uses two queues, namely QUEUE1 and QUEUE2. QUEUE1 holds all the nodes that are to be processed, while QUEUE2 holds all the nodes that are processed and deleted from QUEUE1.

Now, let's start examining the graph starting from Node A.

Step 1 - First, add A to queue1 and NULL to queue2.

1. QUEUE1 = {A}
2. QUEUE2 = {NULL}

Step 2 -
Now, delete node A from queue1 and add it into queue2. Insert all neighbors of node A to queue1.

1. QUEUE1 = {B, D}
2. QUEUE2 = {A}

Step 3 -
Now, delete node B from queue1 and add it into queue2. Insert all neighbors of node B to queue1.

1. QUEUE1 = {D, C, F}
2. QUEUE2 = {A, B}

Step 4 -
Now, delete node D from queue1 and add it into queue2. Insert all neighbors o

f node D to queue1. The only neighbor of Node D is F since it is already inserte d, so it will not be inserted again.

1. QUEUE1 = {C, F}
2. QUEUE2 = {A, B, D}

Step 5 -
Delete node C from queue1 and add it into queue2. Insert all neighbors of nod e C to queue1.

1. QUEUE1 = {F, E, G}
2. QUEUE2 = {A, B, D, C}

Step 5 -
Delete node F from queue1 and add it into queue2. Insert all neighbors of nod e F to queue1. Since all the neighbors of node F are already present, we will not insert them again.

1. QUEUE1 = {E, G}
2. QUEUE2 = {A, B, D, C, F}

Step 6 -
Delete node E from queue1. Since all of its neighbors have already been added , so we will not insert them again. Now, all the nodes are visited, and the target node E is encountered into queue2.

1. QUEUE1 = {G}
2. QUEUE2 = {A, B, D, C, F, E}

Complexity of BFS algorithm

Time complexity of BFS depends upon the data structure used to represent the graph. The time complexity of BFS algorithm is **O(V+E)**, since in the worst case, BFS algorithm explores every node and edge. In a graph, the number of vertice s is O(V), whereas the number of edges is O(E).

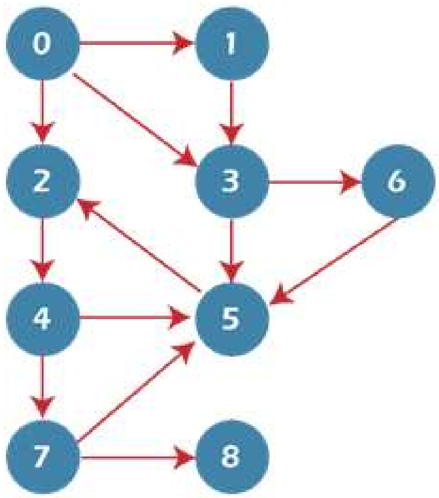
The space complexity of BFS can be expressed as **O(V)**, where V is the number of vertices.

Implementation of BFS algorithm

Now, let's see the implementation of BFS algorithm in java.

In this code, we are using the adjacency list to represent our graph. Implementing the Breadth-First Search algorithm in Java makes it much easier to deal with the adjacency list since we only have to travel through the list of nodes attached to each node once the node is dequeued from the head (or start) of the queue.

In this example, the graph that we are using to demonstrate the code is given as follows -



Spanning tree

In this article, we will discuss the spanning tree and the minimum spanning tree. But before moving directly towards the spanning tree, let's first see a brief description of the graph and its types.

Graph

A graph can be defined as a group of vertices and edges to connect these vertices. The types of graphs are given as follows -

- **Undirected graph:** An undirected graph is a graph in which all the edges do not point to any particular direction, i.e., they are not unidirectional; they are bidirectional. It can also be defined as a graph with a set of V vertices and a set of E edges, each edge connecting two different vertices.
- **Connected graph:** A connected graph is a graph in which a path always exists from a vertex to any other vertex. A graph is connected if we can reach any vertex from any other vertex by following edges in either direction.

- **Directed graph:** Directed graphs are also known as digraphs. A graph is a directed graph (or digraph) if all the edges present between any vertices or nodes of the graph are directed or have a defined direction.

Now, let's move towards the topic spanning tree.

What is a spanning tree?

A spanning tree can be defined as the subgraph of an undirected connected graph. It includes all the vertices along with the least possible number of edges. If any vertex is missed, it is not a spanning tree. A spanning tree is a subset of the graph that does not have cycles, and it also cannot be disconnected.

A spanning tree consists of (n-1) edges, where 'n' is the number of vertices (or nodes). Edges of the spanning tree may or may not have weights assigned to them. All the possible spanning trees created from the given graph G would have the same number of vertices, but the number of edges in the spanning tree would be equal to the number of vertices in the given graph minus 1.

A complete undirected graph can have $\mathbf{n^{n-2}}$ number of spanning trees where **n** is the number of vertices in the graph. Suppose, if **n = 5**, the number of maximum possible spanning trees would be $\mathbf{5^{5-2} = 125}$.

Applications of the spanning tree

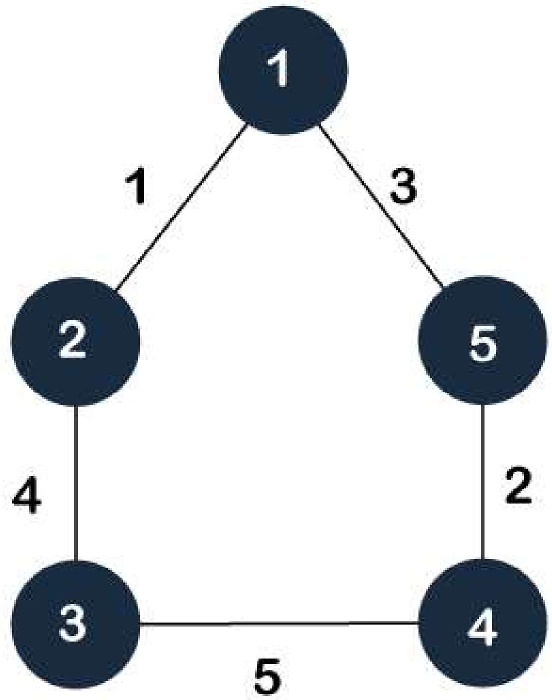
Basically, a spanning tree is used to find a minimum path to connect all nodes of the graph. Some of the common applications of the spanning tree are listed as follows -

- Cluster Analysis
- Civil network planning
- Computer network routing protocol

Now, let's understand the spanning tree with the help of an example.

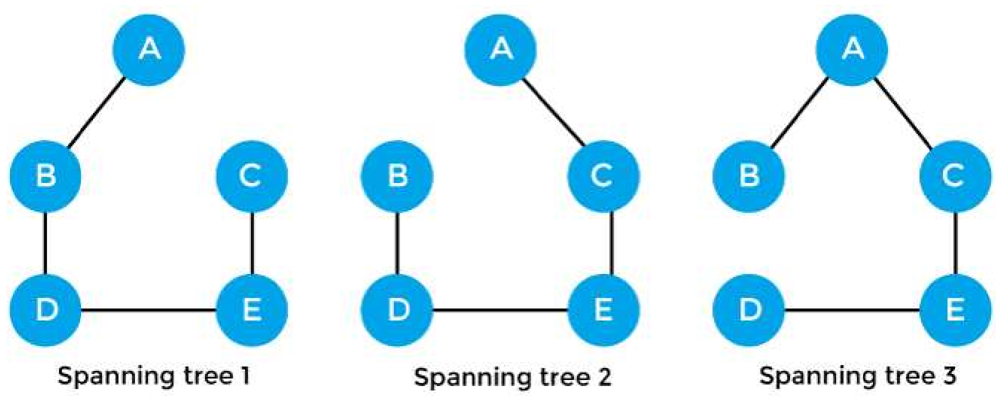
Example of Spanning tree

Suppose the graph be -



As discussed above, a spanning tree contains the same number of vertices as the graph, the number of vertices in the above graph is 5; therefore, the spanning tree will contain 5 vertices. The edges in the spanning tree will be equal to the number of vertices in the graph minus 1. So, there will be 4 edges in the spanning tree.

Some of the possible spanning trees that will be created from the above graph are given as follows -



Properties of spanning-tree

Some of the properties of the spanning tree are given as follows -

- There can be more than one spanning tree of a connected graph G.

- A spanning tree does not have any cycles or loop.
- A spanning tree is **minimally connected,** so removing one edge from the tree will make the graph disconnected.
- A spanning tree is **maximally acyclic,** so adding one edge to the tree will crea te a loop.
- There can be a maximum $\mathbf{n^{n-2}}$ number of spanning trees that can be created from a complete graph.
- A spanning tree has **n-1** edges, where 'n' is the number of nodes.
- If the graph is a complete graph, then the spanning tree can be constructed by removing maximum (e-n+1) edges, where 'e' is the number of edges and 'n' is the number of vertices.

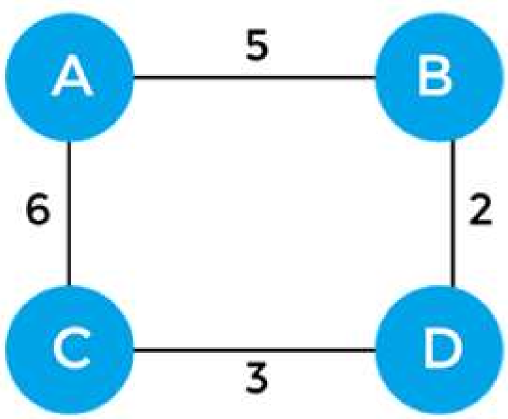
So, a spanning tree is a subset of connected graph G, and there is no spanning tree of a disconnected graph.

Minimum Spanning tree

A minimum spanning tree can be defined as the spanning tree in which the su m of the weights of the edge is minimum. The weight of the spanning tree is th e sum of the weights given to the edges of the spanning tree. In the real world, this weight can be considered as the distance, traffic load, congestion, or any ra ndom value.

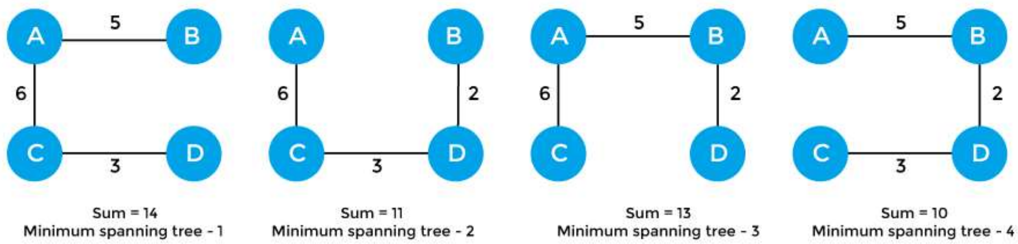
Example of minimum spanning tree

Let's understand the minimum spanning tree with the help of an example.

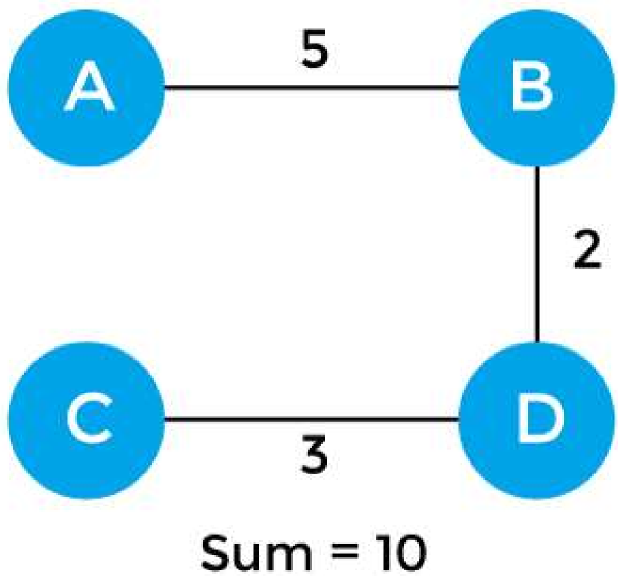


Weighted graph

The sum of the edges of the above graph is 16. Now, some of the possible span ning trees created from the above graph are -



So, the minimum spanning tree that is selected from the above spanning trees f or the given weighted graph is -



Applications of minimum spanning tree

The applications of the minimum spanning tree are given as follows -

- Minimum spanning tree can be used to design water-supply networks, telecommunication networks, and electrical grids.
- It can be used to find paths in the map.

Algorithms for Minimum spanning tree

A minimum spanning tree can be found from a weighted graph by using the al gorithms given below -

- Prim's Algorithm

- Kruskal's Algorithm

Let's see a brief description of both of the algorithms listed above.

Prim's algorithm -
It is a greedy algorithm that starts with an empty spanning tree. It is used to fin d the minimum spanning tree from the graph. This algorithm finds the subset o f edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized.

To learn more about the prim's algorithm, you can click the below link - https://www.javatpoint.com/prim-algorithm

Kruskal's algorithm -
This algorithm is also used to find the minimum spanning tree for a connected weighted graph. Kruskal's algorithm also follows greedy approach, which finds an optimum solution at every stage instead of focusing on a global optimum.

To learn more about the prim's algorithm, you can click the below link - https://www.javatpoint.com/kruskal-algorithm

So, that's all about the article. Hope the article will be helpful and informative t o you. Here, we have discussed spanning tree and minimum spanning tree alon g with their properties, examples, and applications.

vvvvvvvv