Unit 6
Relations and Graphs

6.1 Relations

Relations

- Let A and B be sets. A binary relation from A to B is a subset of A × B.
- In other words, a binary relation from A to B is a set R of ordered pairs where the first element of each ordered pair comes from A and the second element comes from B.
- We use the notation $a\ R\ b$ to denote that $(a, b)\ \in\ R$ and $a \not R b$ to denote that $(a, b) \notin R$. Moreover, when (a, b) belongs to R, a is said to be *related to* b by R.

EXAMPLE: Let A = {0, 1, 2} and B = {a, b}. Then {(0, a), (0, b), (1, a), (2, b)} is a relation from A to B. This means, for instance, that 0 R a, but that $1 \not R b$.

Relations on a Set

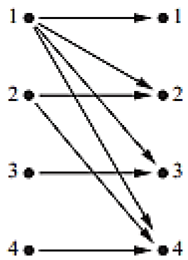
- A relation on a set A is a relation from A to A.
- In other words, a relation on a set A is a subset of A × A

EXAMPLE: Let A be the set {1, 2, 3, 4}. Which ordered pairs are in the relation R = {(a, b) | a divides b}?

Solution: Because (a, b) is in R if and only if a and b are positive integers not exceeding 4 such that a divides b, we see that

R = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)}.

The pairs in this relation are displayed both graphically and in tabular form as:



R	1	2	3	4
1	×	×	×	×
2		×		×
3			×	
4				×

<u>Domain and Range of Relations</u>

Suppose R is a relation from A to B then, the domain of relation R is the set of all first elements of ordered pairs which belongs to R. It is denoted by Dom(R).

Mathematically,

Dom(R) = {a ∈ A / (a, b) ∈ R for some b ∈ B}

Similarly, the range of relation R is the set of all second elements of ordered pairs belongs to relation R. It is denoted by Range(R).

Mathematically,

Range(R) = {b ∈ B / (a, b) ∈ R for some a ∈ A}

EXAMPLE: Let A = {4, 5, 6}, find the relations in A×A under the collection x + y < 10. Also find domain and range of relation.

Solution:

- The relation A x A under the collection x + y < 10 is:
  R = {(4, 4), (4, 5)}
- The domain is:
  Dom(R) = {4}
- The range is:
  Range(R) = {4, 5}

Properties of Relations

There are several properties that are used to classify relations on a set.

Reflexive Relations

- A relation R on a set A is called reflexive if (a, a) ∈ R for every element a ∈ A.

EXAMPLE: Consider the following relations on {1, 2, 3, 4}:

R1 = {(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)},

R2 = {(1, 1), (1, 2), (2, 1)},

R3 = {(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)},

R4 = {(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)},

R5 = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)},

R6 = {(3, 4)}.

Which of these relations are reflexive?

Solution:

- The relations R3 and R5 are reflexive because they both contain all pairs of the form (a, a), namely, (1, 1), (2, 2), (3, 3), and (4, 4).
- The other relations are not reflexive because they do not contain all of these ordered pairs.
- In particular, R1, R2, R4, and R6 are not reflexive because (3, 3) is not in any of these relations.

Symmetric Relations

- A relation R on a set A is called symmetric if (b, a) ∈ R whenever (a, b) ∈ R, for all a, b ∈ A.

EXAMPLE: R = {(1,1), (1,2), (1,3), (2,3), (2,1), (3,2), (3,1)} is a symmetric relation since for (1,2), (1,3), (2,3) there are (2,1), (3,1), and (3,2) respectively.

Asymmetric Relations

- A relation R on a set A is asymmetric if (a, b) ∈ R then (b, a) ∉ R for a, b ∈ A.

EXAMPLE: R = {(1,2), (2,3), (3,1)} on A = {1,2,3} is asymmetric since for (1,2) ∈ R, there is no (2,1) in R. Similarly (3, 2) ∉ R and (1, 3) ∉ R.

Antisymmetric Relations

- A relation R on a set A such that for all a, b ∈ A, if (a, b) ∈ R and (b, a) ∈ R, then a = b is called antisymmetric.
- The contrapositive of this definition is that R is antisymmetric if a $\not R$ b or b $\not R$ a whenever a≠b.

EXAMPLE: R = {(1,2), (2,2), (2,3)} on A = (1,2,3) is an antisymmetric since if we choose 1 and 2 then for 1≠2, (1,2) ∈ R but (2,1) ∉ R. Again if we choose 2 and 3 then for 2≠3 the for (2,3) ∈ R but (3,2) ∉ R.

Transitive Relations

- A relation R on a set A is called transitive if whenever (a, b) ∈ R and (b, c) ∈ R, then (a, c) ∈ R, for all a, b, c ∈ A.

EXAMPLE: R4, R5, and R6 are transitive from the above example.

Combining Relations

Because relations from A to B are subsets of A × B, two relations from A to B can be combined in any way two sets can be combined.

EXAMPLE: Let A = {1, 2, 3} and B = {1, 2, 3, 4}. The relations R1 = {(1, 1), (2, 2), (3, 3)} and R2 = {(1, 1), (1, 2), (1, 3), (1, 4)} can be combined to obtain

R1 ∪ R2 = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)},

R1 ∩ R2 = {(1, 1)},

R1 − R2 = {(2, 2), (3, 3)},

R2 − R1 = {(1, 2), (1, 3), (1, 4)}.

N-ary Relations and their applications

n-ary Relations

- Let $A_1, A_2, \ldots, A_n$ be sets.
- An n-ary relation on these sets is a subset of $A_1 \times A_2 \times \cdots \times A_n$.
- The sets $A_1, A_2, \ldots, A_n$ are called the domains of the relation, and n is called its degree.

EXAMPLE:

- Let R be the relation on N × N × N consisting of triples (a, b, c), where a, b, and c are integers with a<b<c.
- Then (1, 2, 3) ∈ R, but (2, 4, 3) ∉ R.
- The degree of this relation is 3.
- Its domains are all equal to the set of natural numbers.

Applications of n-ary Relations(Database)

- A database consists of records, which are n-tuples, made up of fields.
- The fields are the entries of the n-tuples.
- For instance, a database of student records may be made up of fields containing the name, student number, major, and grade point average of the student.
- The relational data model represents a database of records as an n-ary relation.
- Thus, student records are represented as 4-tuples of the form *(Student_name, ID_number, Major, GPA)*. A sample database of six such records is

Student Table

Student_name	***ID_number***	***Major***	***GPA***
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Tom	453876	Mathematics	3.45
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Primary Key

A domain of an n-ary relation is called a primary key when the value of the n-tuple from this domain determines the n-tuple.

Composite Key

Combinations of domains can also uniquely identify n-tuples in an n-ary relation. When the values of a set of domains determine an n-tuple in a relation, the Cartesian product of these domains is called a composite key.

Operations on n-ary Relations

Selection

Let R be an n-ary relation and C a condition that elements in R may satisfy. Then the selection operator S_C maps the n-ary relation R to the n-ary relation of all n-tuples from R that satisfy the condition C.

EXAMPLE: To find the records of computer science majors in the n-ary relation R shown in Student Table.

Student Table

Student_name	***ID_number***	***Major***	***GPA***
Ackermann	231455	Computer Science	3.88
Chou	102147	Computer Science	3.49

Projection

The projection $P_{i_1, i_2, \ldots\ldots, i_m}$ where $i_1 < i_2 < \cdots < i_m$, maps the n-tuple $(a_1, a_2, \ldots, a_n)$ to the m-tuple $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$, where $m \leq n$.

EXAMPLE: What relation results when the projection $P_{1,4}$ is applied to the relation in Student Table?

Student Table

Student_name	***GPA***
Ackermann	3.88
Adams	3.45
Chou	3.49
Tom	3.45
Rao	3.90
Stevens	2.99

Join

Let R be a relation of degree m and S a relation of degree n. The join $J_p(R, S)$, where $p \leq m$ and $p \leq n$, is a relation of degree $m + n - p$ that consists of all $(m + n - p)$-tuples$(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b2, \ldots, b_{n-p})$, where the m-tuple $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ belongs to R and the n-tuple $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ belongs to S.

EXAMPLE: What relation results when the join operator *J_p(Student, Department)* is used to combine the relation displayed in Student Tables and Department Table?

Student Table

Student_name	***ID_number***	***Dep_Name***	***GPA***
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Tom	453876	Mathematics	3.45
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Department Table

Dep_ID	*Dep_Name*
01	Computer Science
02	Physics
03	Mathematics
04	Psychology

Solution:

- The join J_p*(Student, Department)* produces the relation shown in Table below.

Student_name	*ID_number*	*Dep_Name*	*GPA*	*Dep_ID*
Ackermann	231455	Computer Science	3.88	01
Adams	888323	Physics	3.45	02
Chou	102147	Computer Science	3.49	01
Tom	453876	Mathematics	3.45	03
Rao	678543	Mathematics	3.90	03
Stevens	786576	Psychology	2.99	04

Representing Relations

Matrix Representation of Relations

- A relation between finite sets can be represented using a zero–one matrix.
- Suppose that R is a relation from A = $\{a_1, a_2, ..., a_m\}$ to B = $\{b_1, b_2, ..., b_n\}$. The relation R can be represented by the matrix $M_R = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 \text{ if } (a_i, b_j) \in R, \\ 0 \text{ if } (a_i, b_j) \notin R. \end{cases}$$

EXAMPLE: Suppose that A = {1, 2, 3} and B = {1, 2}. Let R be the relation from A to B containing (a, b) if a ∈ A, b ∈ B, and a>b. What is the matrix representing R if $a_1 = 1$, $a_2 = 2$, and $a_3 = 3$, and $b_1 = 1$ and $b_2 = 2$?

Solution:

- Because R = {(2, 1), (3, 1), (3, 2)}, the matrix for R is

$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

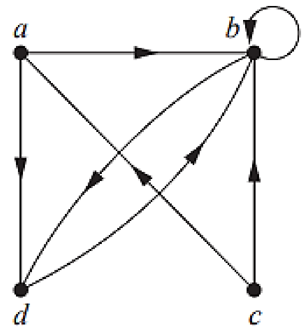
- The 1s in MR show that the pairs (2, 1), (3, 1), and (3, 2) belong to R. The 0s show that no other pairs belong to R.

Representing Relations Using Digraphs

- A directed graph, or digraph, consists of a set V of vertices (or nodes) together with a set E of ordered pairs of elements of V called edges (or arcs).

- The vertex a is called the initial vertex of the edge (a, b), and the vertex b is called the terminal vertex of this edge.

EXAMPLE: The directed graph with vertices a, b, c, and d, and edges (a, b), (a, d), (b, b), (b, d), (c, a), (c, b), and (d, b) is displayed in Figure below.



Reflexive Closure of Relations

- Given a relation R on a set A, the reflexive closure of R can be formed by adding to R all pairs of the form (a, a) with $a \in A$, not already in R.
- The addition of these pairs produces a new relation that is reflexive, contains R, and is contained within any reflexive relation containing R.
- We see that the reflexive closure of R equals $R \cup \Delta$, where $\Delta = \{(a, a) \mid a \in A\}$ is the diagonal relation on A.

The reflexive closure of R is

$$R \cup \Delta = \{(a, b) \mid a < b\} \cup \{(a, a) \mid a \in Z\} = \{(a, b) \mid a \leq b\}.$$

Symmetric Closure of Relations

The symmetric closure of R is the relation

$$R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\} = \{(a, b) \mid a \neq b\}$$

Transitive Closures

- Let R be a relation on a set A. There is a path of length n, where n is a positive integer, from a to b if and only if $(a, b) \in R^n$.
- Let R be a relation on a set A. The connectivity relation R^* consists of the pairs (a, b) such that there is a path of length at least one from a to b in R.
- Because R_n consists of the pairs (a, b) such that there is a path of length n from a to b, it follows that R^* is the union of all the sets R^n.
- In other words,

$$R^* = \bigcup_{n=1}^{\infty} R^n.$$

- The transitive closure of a relation R equals the connectivity relation R^*.

Equivalence Relations

A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

Equivalence Classes

- Let R be an equivalence relation on a set A. The set of all elements that are related to an element a of A is called the equivalence class of a.
- The equivalence class of a with respect to R is denoted by $[a]_R$.
- In other words, if R is an equivalence relation on a set A, the equivalence class of the element a is

  $[a]_R = \{s \mid (a, s) \in R\}$.

- If $b \in [a]_R$, then b is called a representative of this equivalence class.
- Any element of a class can be used as a representative of this class.
- That is, there is nothing special about the particular element chosen as the representative of the class.

EXAMPLE: What are the equivalence classes of 0 for congruence modulo 4?

Solution:

- The equivalence class of 0 contains all integers a such that $a \equiv 0 \pmod 4$.
- The integers in this class are those divisible by 4.
- Hence, the equivalence class of 0 for this relation is

  $[0] = \{\ldots, -8, -4, 0, 4, 8, \ldots\}$

Exercise

1. Find the equivalence classes of 1, 2, and 3 for set A = {1, 2, 3, 4, 5} and R = {(a,b) | a+b is even} such that R is defined on set A.

Partitions

- A partition of a set S is a collection of disjoint nonempty subsets of S that have S as their union.

Partial Ordering

- A relation R on a set S is called a partial ordering or partial order if it is reflexive, antisymmetric, and transitive.
- A set S together with a partial ordering R is called a partially ordered set, or poset, and is denoted by (S, R).
- Members of S are called elements of the poset.

EXAMPLE: Show that the "greater than or equal" relation (≥) is a partial ordering on the set of integers.

Solution:

- Because $a \geq a$ for every integer a, ≥ is reflexive.
- If $a \geq b$ and $b \geq a$, then $a = b$.
- Hence, ≥ is antisymmetric.
- Finally, ≥ is transitive because $a \geq b$ and $b \geq c$ imply that $a \geq c$.
- It follows that ≥ is a partial ordering on the set of integers and (Z, ≥) is a poset.

Total Ordering

- The elements a and b of a poset $(S, \preccurlyeq)$ are called comparable if either $a \preccurlyeq b$ or $b \preccurlyeq a$. When a and b are elements of S such that neither $a \preccurlyeq b$ nor $b \preccurlyeq a$, a and b are called incomparable.
- If $(S, \preccurlyeq)$ is a poset and every two elements of S are comparable, S is called a totally ordered or linearly ordered set, and $\preccurlyeq$ is called a total order or a linear order.
- A totally ordered set is also called a chain.
- $(S, \preccurlyeq)$ is a well-ordered set if it is a poset such that is a total ordering and every nonempty subset of S has a least element.

EXAMPLE:

- If A is any set of positive integer (A, /) is not TOS.
- If A is any set of positive integer (A, ≤) is TOS.

Lexicographical ordering

- The words in a dictionary are listed in alphabetic, or lexicographic, order, which is based on the ordering of the letters in the alphabet.
- This is a special case of an ordering of strings on a set constructed from a partial ordering on the set.

Lattice

- A partially ordered set in which every pair of elements has both a least upper bound and a greatest lower bound is called a lattice.
- Lattices have many special properties.

- Furthermore, lattices are used in many different applications such as models of information flow and play an important role in Boolean algebra.

Hasse Diagrams

- A Hasse diagram is a graphical representation of the relation of elements of a partially ordered set (poset) with an implied upward orientation.

EXAMPLE: Draw the Hasse diagram representing the partial ordering {(a, b)|a divides b} on

{1, 2, 3, 4, 6, 8, 12}.

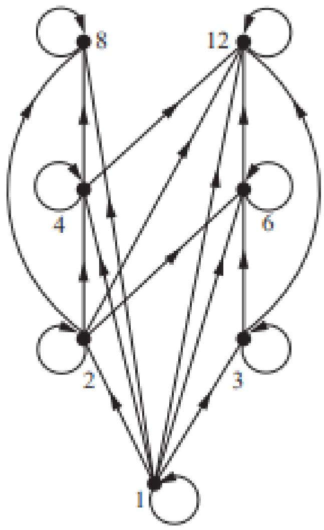
Solution:

Here,

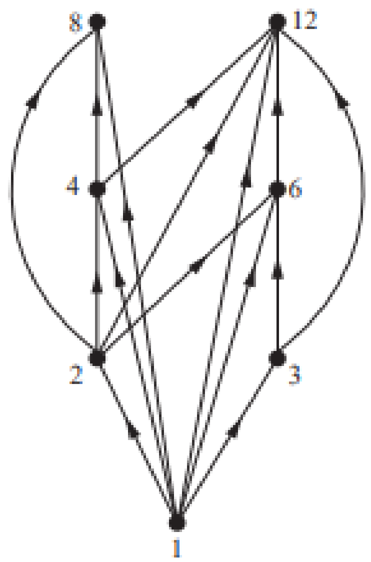
A={1, 2, 3, 4, 6, 8, 12}

R={(1,1), (1,2), (1,3), (1,4), (1,6), (1,8), (1,12), (2,2), (2,4), (2,6), (2,8), (2,12), (3,3), (3,6), (3,12), (4,4), (4,8), (4,12), (6,6), (6,12), (8,8), (12,12)}.

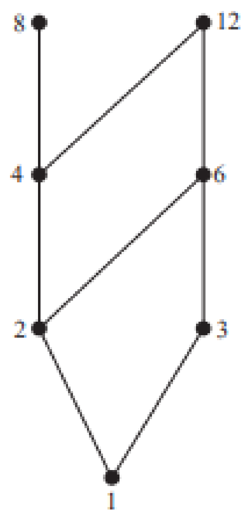
Step 1: We construct a directed graph corresponding a relation R.



Step 2: We remove all loops from the diagram (reflexivity) and all transitive edges.



Step 3: We make sure that the initial vertex is below the terminal vertex and remove all arrows.

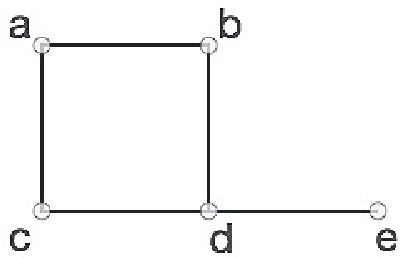


6.2 Graphs

Graphs Definition

- A graph G = (V, E) consists of V, a nonempty set of vertices (or nodes) and E, a set of edges.
- Each edge has either one or two vertices associated with it, called its endpoints.
- An edge is said to connect its endpoints.
- The set of vertices V of a graph G may be infinite.
- A graph with an infinite vertex set or an infinite number of edges is called an *infinite* graph, and in comparison, a graph with a finite vertex set and a finite edge set is called a *finite graph*.

EXAMPLE:



In the above graph,

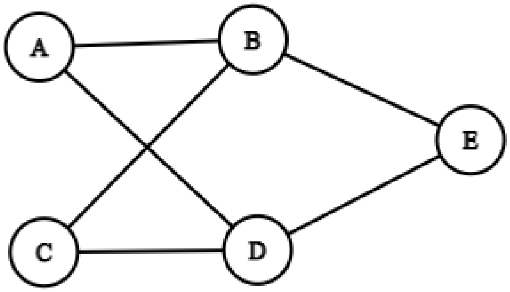
V = {a, b, c, d, e}

E = {(a, b), (a, c), (b, d), (c, d), (d, e)}

Types of Graphs

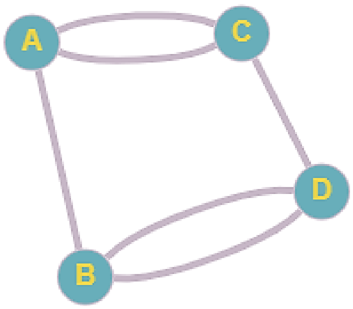
Simple Graphs

- A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a *simple graph*.



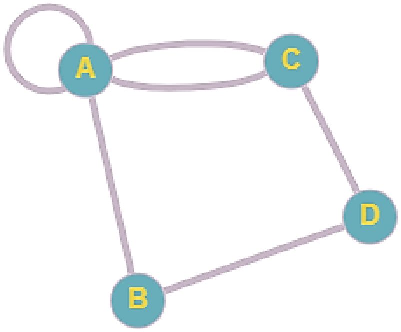
Multigraphs

- Graphs that may have multiple edges connecting the same vertices are called multigraphs.



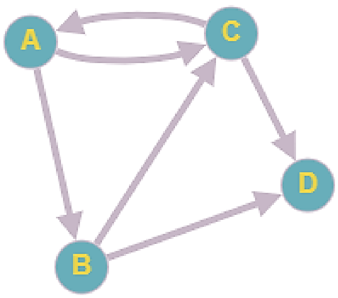
Pseudographs

- The edges that connect a vertex to itself, such edges are called *loops.*
- Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called pseudographs.



Directed Graphs

- A directed graph (or digraph) (V, E) consists of a nonempty set of vertices V and a set of directed edges (or arcs) E.
- Each directed edge is associated with an ordered pair of vertices.
- The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.



Simple Directed Graph

- When a directed graph has no loops and has no multiple directed edges, it is called a simple directed graph.

Directed Multigraph

- Directed graphs that may have multiple directed edges from a vertex to a second (possibly the same) vertex are called directed multigraphs.
- When there are m directed edges, each associated to an ordered pair of vertices (u, v), we say that (u, v) is an edge of multiplicity m.

Mixed Graph

- A graph with both directed and undirected edges is called a mixed graph.

Graph Models

Graphs are used in a wide variety of models.

- Social Networks
- Communication Networks
- Information Networks

- Software Design Applications
- Transportation Network
- Biological Networks and
- Tournaments

Graph Terminologies

Adjacent Vertices and Incident Edges

- Two vertices u and v in an undirected graph G are called *adjacent* (or neighbours) in G if u and v are endpoints of an edge e of G.
- Such an edge e is called *incident* with the vertices u and v and e is said to connect u and v.

Neighbourhood Vertices

- The set of all neighbours of a vertex v of G = (V, E), denoted by N (v), is called the neighbourhood of v.
- If A is a subset of V, we denote by N (A) the set of all vertices in G that are adjacent to at least one vertex in A.
- So, $N\ (A)\ =\ \bigcup_{v\in A} N\ (v)$.

Degree of a Vertix

- The degree of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.
- The degree of the vertex v is denoted by deg(v).

EXAMPLE: What are the degrees and what are the neighbourhoods of the vertices in the graphs G and H displayed in figure below.

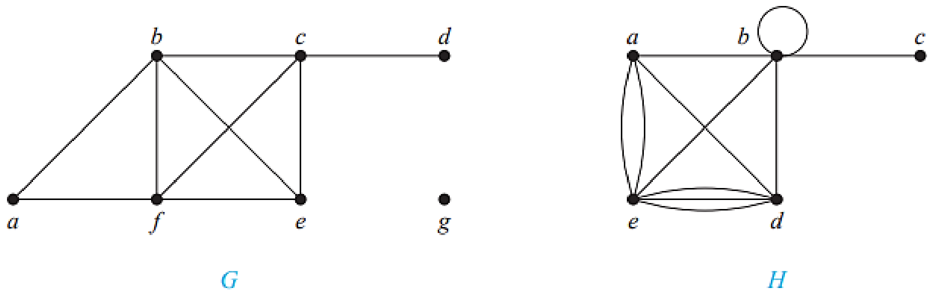


Fig: The Undirected Graphs G and H.

Solution:

- In G, deg(a) = 2, deg(b) = deg(c) = deg(f) = 4, deg(d) = 1, deg(e) = 3, and deg(g) = 0.
- The neighbourhoods of these vertices are N (a) = {b, f }, N (b) = {a, c, e, f }, N (c) = {b, d, e, f }, N (d) = {c}, N (e) = {b, c, f }, N (f) = {a, b, c, e}, and N (g) = ∅.
- In H, deg(a) = 4, deg(b) = deg(e) = 6, deg(c) = 1, and deg(d) = 5.
- The neighbourhoods of these vertices are N (a) = {b, d, e}, N (b) = {a, b, c, d, e}, N (c) = {b}, N (d) = {a, b, e}, and N (e) = {a, b, d}.

Isolated and Pendant Vertices

- A vertex of degree zero is called *isolated*.
- It follows that an isolated vertex is not adjacent to any vertex.
- Vertex g in graph G in Example above is isolated.
- A vertex is *pendant* if and only if it has degree one.
- Consequently, a pendant vertex is adjacent to exactly one other vertex.
- Vertex d in graph G in Example above is pendant.

In-degree and Out-degree

- In a graph with directed edges the in-degree of a vertex v, denoted by deg−(v), is the number of edges with v as their terminal vertex.
- The out-degree of v, denoted by deg+(v), is the number of edges with v as their initial vertex.
- (Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

EXAMPLE: Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in Figure below.

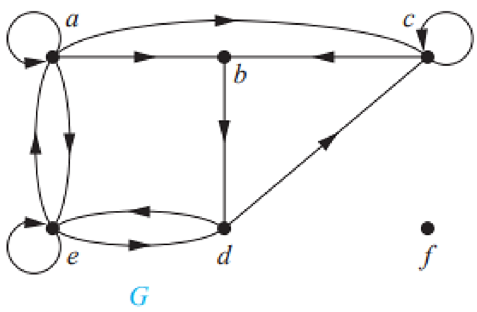


Fig: The Directed Graph G.

Solution:

- The in-degrees in G are deg−(a) = 2, deg−(b) = 2, deg−(c) = 3, deg−(d) = 2, deg−(e) = 3, and deg−(f) = 0.
- The out-degrees are deg+(a) = 4, deg+(b) = 1, deg+(c) = 2, deg+(d) = 2, deg+(e) = 3, and deg+(f) = 0.

Underlying undirected graph

- The undirected graph that results from ignoring directions of edges is called the underlying undirected graph.

Some Special Simple Graphs

Complete Graphs

- A complete graph on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.
- The graphs K_n, for n = 1, 2, 3, 4, and 5 are displayed in Figure below.
- A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called non-complete.

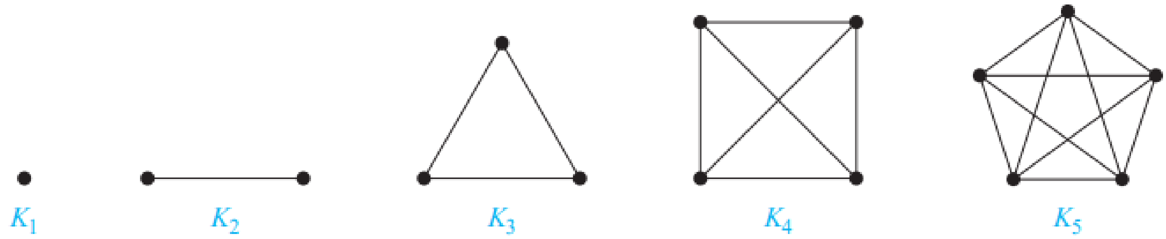


Fig: The Graphs K_n for $1 \leq n \leq 5$.

Cycles

- A cycle C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \dots, v_n$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$.

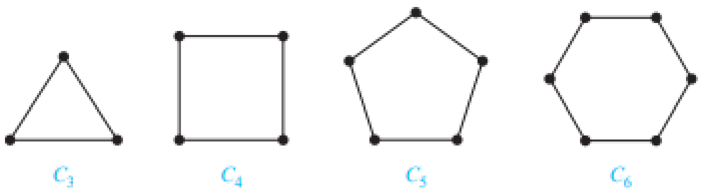


Fig: The Cycles C_3, C_4, C_5, and C_6.

Wheels

We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges.

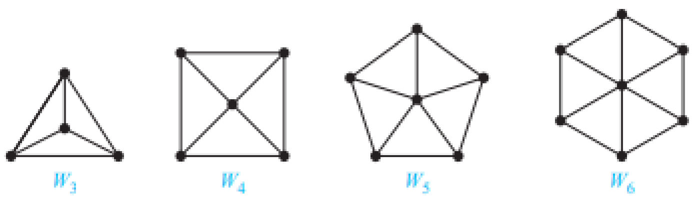


Fig: The Wheels W_3, W_4, W_5, and W_6.

n-Cubes

- An n-dimensional hypercube, or n-cube, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length n.
- Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position.

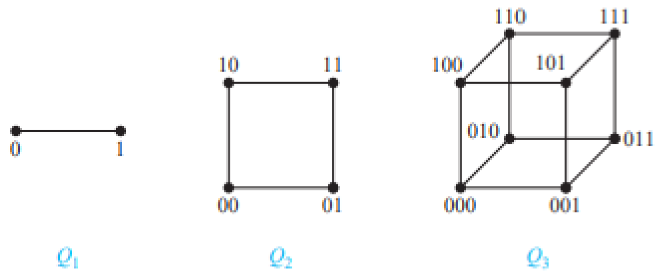
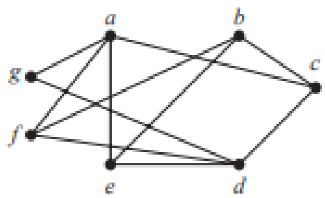


Fig: The n-cube Q_n, n = 1, 2, 3

Bipartite Graphs

- A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2).
- When this condition holds, we call the pair (V_1, V_2) a bipartition of the vertex set V of G.

EXAMPLE: Is the graph displayed in figure bipartite?



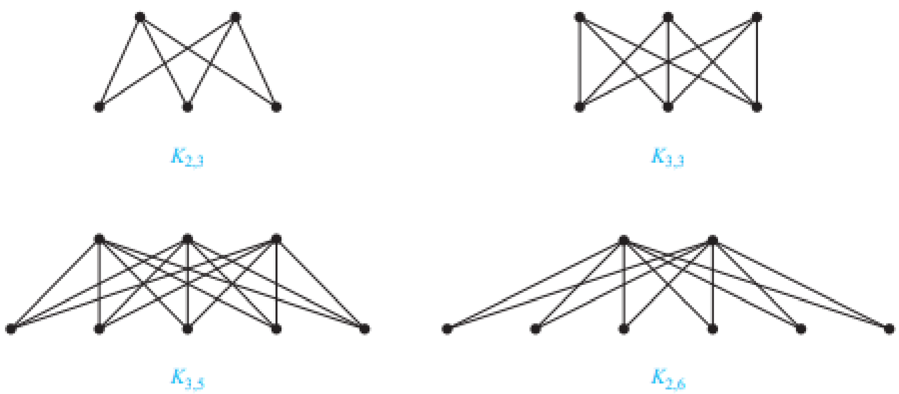
Solution:

- Graph G is bipartite because its vertex set is the union of two disjoint sets, {a, b, d} and {c, e, f, g}, and each edge connects a vertex in one of these subsets to a vertex in the other subset.

Complete Bipartite Graphs

- A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.

EXAMPLE:



Subgraph

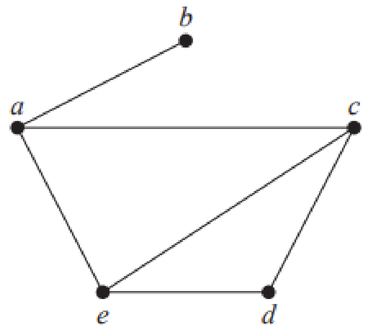
- A subgraph of a graph G = (V, E) is a graph H = (W, F), where W ⊆ V and F ⊆ E. A subgraph H of G is a proper subgraph of G if H = G.
- Let G = (V, E) be a simple graph. The subgraph induced by a subset W of the vertex set V is the graph (W, F), where the edge set F contains an edge in E if and only if both endpoints of this edge are in W.
- The union of two simple graphs G1 = (V1, E1) and G2 = (V2, E2) is the simple graph with vertex set V1 ∪ V2 and edge set E1 ∪ E2. The union of G1 and G2 is denoted by G1 ∪ G2.

Representing Graphs

Adjacency lists

- Specify the vertices that are adjacent to each vertex of the graph.

EXAMPLE:



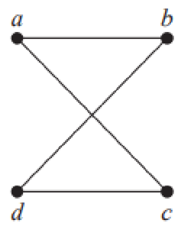
Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

Adjacency Matrices

- The adjacency matrix A (or AG) of G, with respect to this listing of the vertices, is the n x n zero–one matrix with 1 as its $(i, j)^{th}$ entry when v_i and v_j are adjacent, and 0 as its $(i, j)^{th}$ entry when they are not adjacent.
- In other words, if its adjacency matrix is A = $[a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE: Draw a graph with the adjacency matrix



$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Incidence Matrices

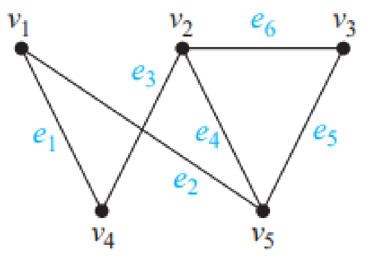
- Another common way to represent graphs is to use incidence matrices.

- Let G = (V, E) be an undirected graph.
- Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G.
- Then the incidence matrix with respect to this ordering of V and E is the n × m matrix M = $[m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE: Represent the graph shown in figure below with an incidence matrix.

Solution: The incidence matrix is

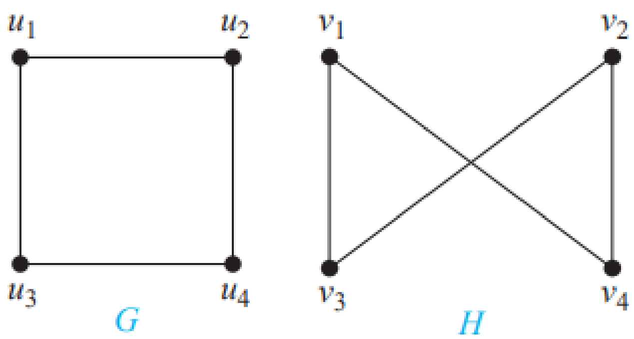


$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Isomorphism of Graphs

- The simple graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic if there exists a one-to-one and onto function f from V1 to V2 with the property that a and b are adjacent in G1 if and only if f(a) and f(b) are adjacent in G2, for all a and b in V1.
- Such a function f is called an isomorphism.
- Two simple graphs that are not isomorphic are called nonisomorphic.

EXAMPLE: Show that the graphs G = (V, E) and H = (W, F), displayed in Figure below, are isomorphic.



Solution:

- The function f with $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$ is a one-to-one correspondence between V and W.

- To see that this correspondence preserves adjacency, note that adjacent vertices in G are u1 and u_2, u_1 and u_3, u_2 and u_4, and u_3 and u_4, and each of the pairs f (u_1) = v_1 and f (u_2) = v_4, f (u_1) = v_1 and f (u_3) = v_3, f (u_2) = v_4 and f (u_4) = v_2, and f (u_3) = v_3 and f (u_4) = v_2 consists of two adjacent vertices in H.

Path

- Let n be a nonnegative integer and G an undirected graph.
- A path of length n from u to v in G is a sequence of n edges $e_1,...,e_n$ of G for which there exists a sequence $x_0 = u, x_1,...,x_{n-1}, x_n = v$ of vertices such that e_i has, for i = 1,...,n, the end points x_{i-1} and x_i.
- When the graph is simple, we denote this path by its vertex sequence $x_0, x_1,...,x_n$ (because listing these vertices uniquely determines the path).
- The path is a circuit if it begins and ends at the same vertex, that is, if u = v, and has length greater than zero.
- The path or circuit is said to pass through the vertices $x_1, x_2,...,x_{n-1}$ or traverse the edges $e_1, e_2,...,e_n$.
- A path or circuit is simple if it does not contain the same edge more than once.

Connectedness in Undirected Graphs

- An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph.
- An undirected graph that is not connected is called disconnected.
- We say that we disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Connected Components

- A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is not connected has two or more connected components that are disjoint and have G as their union.

Connectedness in Directed Graphs

- A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

- A directed graph is weakly connected if there is a path between every two vertices in the underlying undirected graph.

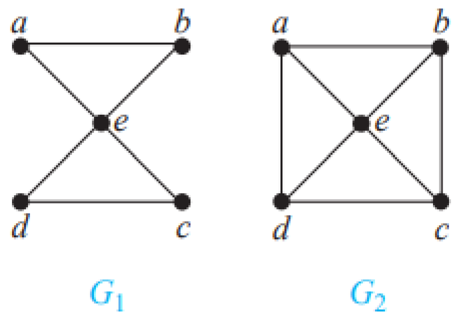
Strong Components of a Directed Graph

- The subgraphs of a directed graph G that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the strongly connected components or strong components of G.
- Note that if a and b are two vertices in a directed graph, their strong components are either the same or disjoint.

Euler Paths and Circuits

- An Euler circuit in a graph G is a simple circuit containing every edge of G.
- An Euler path in G is a simple path containing every edge of G.

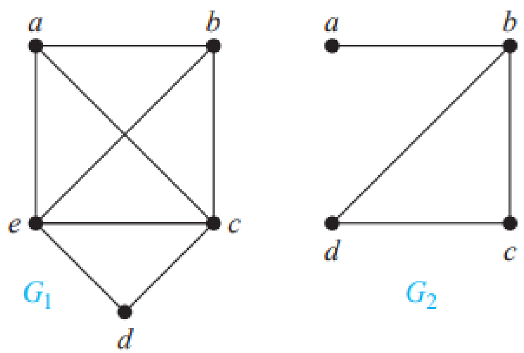
EXAMPLE: Check whether the given graph is Euler Graph or not.



Hamilton Paths and Circuits

- A simple path in a graph G that passes through every vertex exactly once is called a Hamilton path, and a simple circuit in a graph G that passes through every vertex exactly once is called a Hamilton circuit.

EXAMPLE: Check whether the given graph is Hamilton graph or not.



Shortest-Path Problems

Many problems can be modelled using graphs with weights assigned to their edges. Such as:

- Problems involving distances can be modelled by assigning distances between cities to the edges.
- Problems involving flight time can be modelled by assigning flight times to edges.

Weighted Graphs

- Graphs that have a number assigned to each edge are called weighted graphs.
- Weighted graphs are used to model computer networks.
- Communications costs (such as the monthly cost of leasing a telephone line), the response times of the computers over these lines, or the distance between computers, can all be studied using weighted graphs.

EXAMPLE:

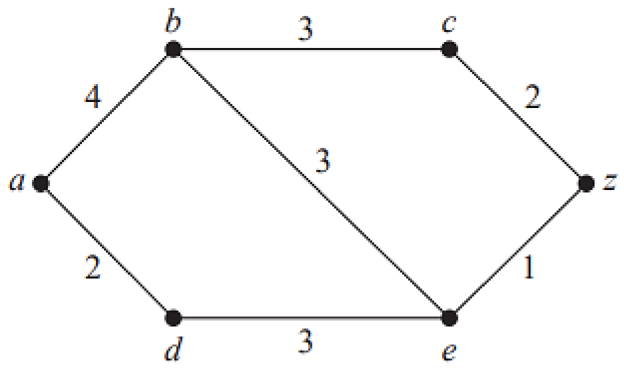


Fig: A Weighted Simple Graph.

Dijkstra's Algorithm

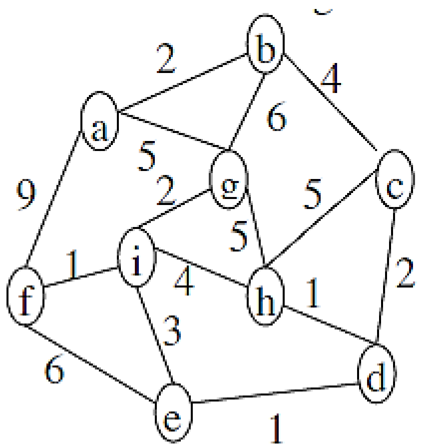
- This approach is of getting single source shortest paths.
- In this algorithm it is assumed that there is no negative weight edge.
- Dijkstra's algorithm works using greedy approach.

Algorithm:

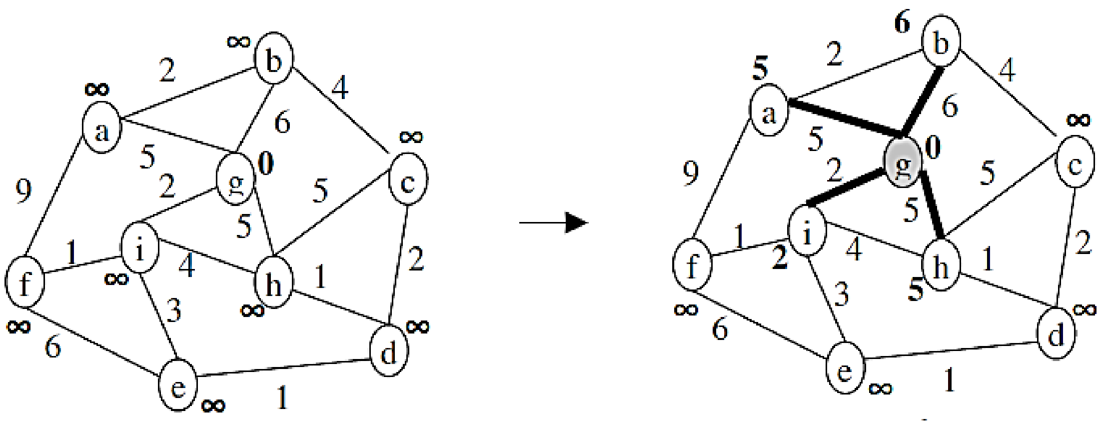
Dijkstra(G,w,s)

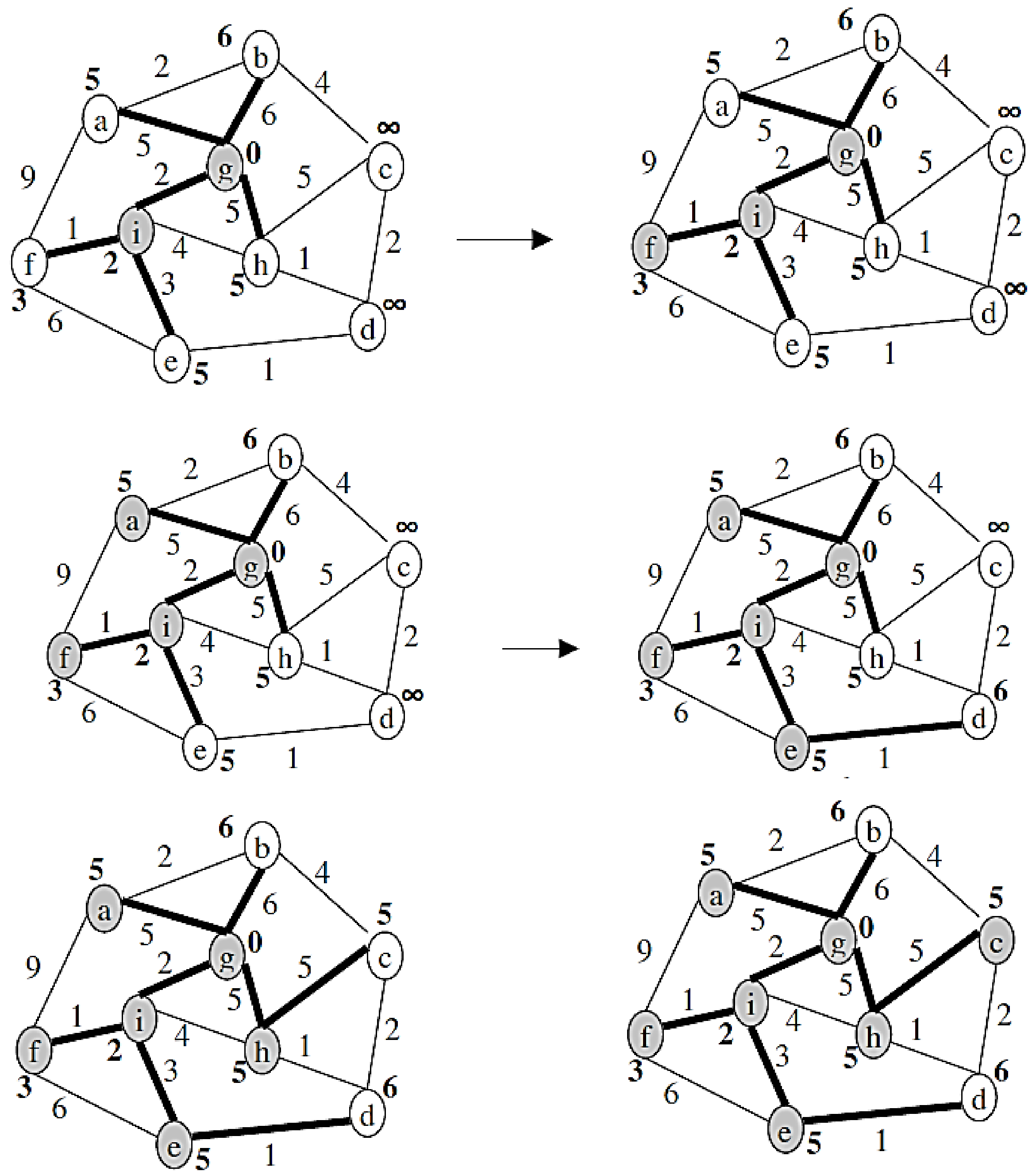
```
{
    for each vertex v∈ V
        do d[v] = ∞
    d[s] = 0
    S = ∅
    Q = V
    While(Q!= ∅)
    {
        u = Take minimum from Q and delete.
        S = S ∪ {u}
        for each vertex v adjacent to u
            do if d[v] > d[u] + w(u,v)
                then d[v] = d[u] + w(u,v)
    }
}
```

EXAMPLE: Find the shortest paths from the source g to all other vertices using Dijkstra's algorithm.



Solution:





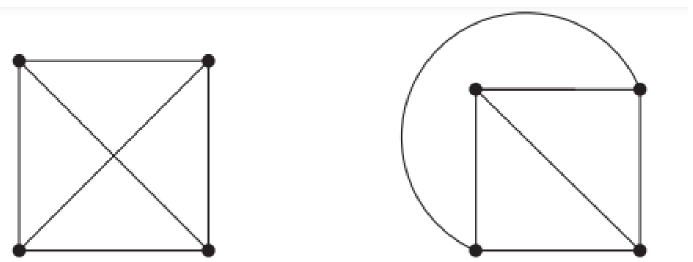
<u>The Traveling Salesperson Problem</u>

A traveling salesperson wants to visit each of n cities exactly once and return to his starting point.

Planar Graphs

- A graph is called planar if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint).
- Such a drawing is called a planar representation of the graph.

EXAMPLE:



Graph Colouring

- A colouring of a simple graph is the assignment of a colour to each vertex of the graph so that no two adjacent vertices are assigned the same colour.

Chromatic number

- The chromatic number of a graph is the least number of colours needed for a colouring of this graph.
- The chromatic number of a graph G is denoted by χ (G). (Here χ is the Greek letter chi.)

Applications of Graph Colorings

Scheduling Final Exams

- This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between two vertices if there is a common student in the courses they represent.
- Each time slot for a final exam is represented by a different color.
- A scheduling of the exams corresponds to a coloring of the associated graph.

EXAMPLE:

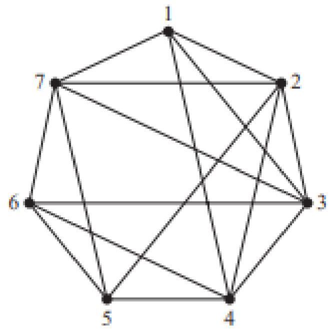
How can the final exams at a university be scheduled so that no student has two exams at the same time?

Solution:

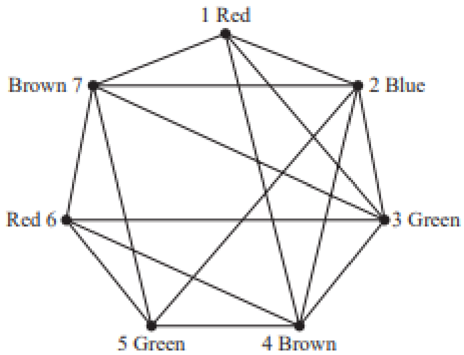
- For instance, suppose there are seven finals to be scheduled.
- Suppose the courses are numbered 1 through 7.
- Suppose that the following pairs of courses have common students: 1 and 2, 1 and 3, 1 and 4, 1 and 7, 2 and 3, 2 and 4, 2 and 5, 2 and 7, 3 and 4, 3 and 6, 3 and 7, 4 and 5, 4 and 6, 5 and 6, 5 and 7, and 6 and 7.
- In figure below the graph associated with this set of classes is shown.
- A scheduling consists of a coloring of this graph. Because the chromatic number of this graph is 4, four time slots are needed.

- A coloring of the graph using four colors and the associated schedule are shown in figure below.

Graph Representing Scheduling of Final Exam



Using Coloring to Schedule Final Exam



Schedule of Final Exam

Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

THEOREM: THE FOUR COLOR THEOREM

The chromatic number of a planar graph is no greater than four.

6.3 Trees

Definition of Tree

A tree is a connected undirected graph with no simple circuits.

Tree Terminologies

Suppose that T is a rooted tree.

Root

- The root node is the topmost node in the tree hierarchy.

Parent and Child

- If v is a vertex in T other than the root, the parent of v is the unique vertex u such that there is a directed edge from u to v.
- When u is the parent of v, v is called a child of u.

Siblings

- Vertices with the same parent are called siblings.

Ancestors

- The ancestors of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root (that is, its parent, its parent's parent, and so on, until the root is reached).

Descendants

- The descendants of a vertex v are those vertices that have v as an ancestor.

Leaf

- A vertex of a rooted tree is called a leaf if it has no children.

Internal vertices

- Vertices that have children are called internal vertices.

Subtree

- If 'a' is a vertex in a tree, the subtree with a as its root is the subgraph of the tree consisting of 'a' and its descendants and all edges incident to these descendants.

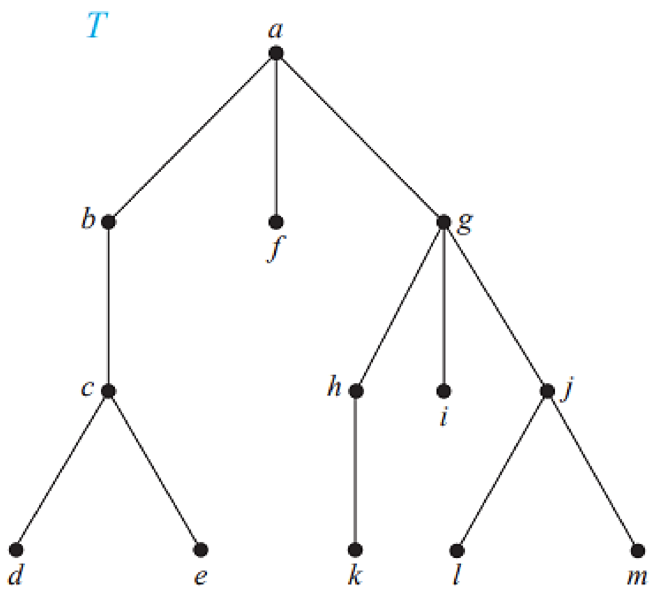


Fig: A tree T with root a.

EXAMPLE: In the rooted tree T (with root a) shown in Figure above, find the parent of c, the children of g, the siblings of h, all ancestors of e, all descendants of b, all internal vertices, and all leaves. What is the subtree rooted at g?

Solution:

- The *parent* of c is b.
- The *children* of g are h, i, and j.
- The *siblings* of h are i and j.
- The *ancestors* of e are c, b, and a.
- The *descendants* of b are c, d, and e.
- The *internal vertices* are a, b, c, g, h, and j.
- The *leaves* are d, e, f, i, k, l, and m.
- The subtree rooted at g is shown in Figure below.

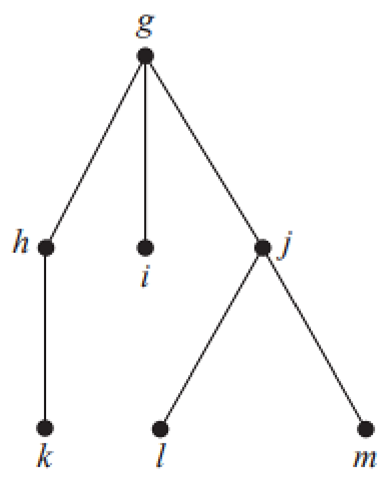


Fig: A subtree with root g.

Applications of trees

The following are the applications of trees:

- *Storing naturally hierarchical data:* Trees are used to store the data in the hierarchical structure. For example, the file system. The file system stored on the disc drive, the file and folder are in the form of the naturally hierarchical data and stored in the form of trees.
- *Organize data:* It is used to organize data for efficient insertion, deletion and searching. For example, a binary tree has a logN time for searching an element.
- *Trie:* It is a special kind of tree that is used to store the dictionary. It is a fast and efficient way for dynamic spell checking.
- *Heap:* It is also a tree data structure implemented using arrays. It is used to implement priority queues.
- *B tree and B+ tree:* B tree and B+ tree are the tree data structures used to implement indexing in databases.
- *Routing table:* The tree data structure is also used to store the data in routing tables in the routers.

Tree Traversals

- The tree traversal is a way in which each node in the tree is visited exactly once in a symmetric manner.
- There are three popular methods of traversal
  - Pre-order traversal
  - In-order traversal
  - Post-order traversal

Pre-order traversal

- The preorder traversal of a nonempty binary tree is defined as follows:
    - Visit the root node
    - Traverse the left sub-tree in preorder
    - Traverse the right sub-tree in preorder
- The preorder is also known as depth first order.

EXAMPLE:

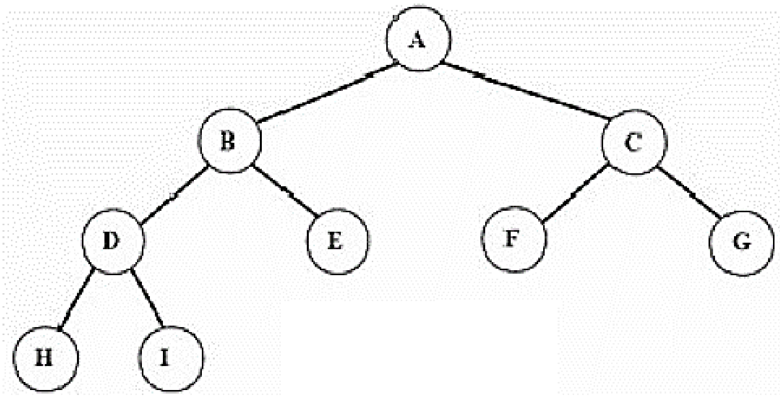


Fig: A binary tree.

- The preorder traversal of given tree is: ABDHIECFG

In-order traversal

- The inorder traversal of a nonempty binary tree is defined as follows:
    - Traverse the left sub-tree in inorder
    - Visit the root node
    - Traverse the right sub-tree in inorder

EXAMPLE:

- The inorder traversal output of the given tree is: H D I B E A F C G

Post-order traversal

- The post-order traversal of a nonempty binary tree is defined as follows:
    - Traverse the left sub-tree in post-order
    - Traverse the right sub-tree in post-order
    - Visit the root node

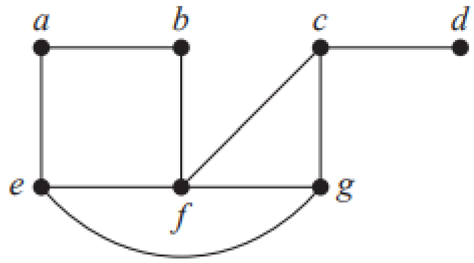
EXAMPLE:

- The post-order traversal output of the given tree is: H I D E B F G C A

Spanning Tree

- Let G be a simple graph.
- A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

EXAMPLE: Find a spanning tree of the simple graph G shown in Figure below.



Solution:

- The graph G is connected, but it is not a tree because it contains simple circuits.
- Remove the edge {a, e}.
- This eliminates one simple circuit, and the resulting subgraph is still connected and still contains every vertex of G.
- Next remove the edge {e, f} to eliminate a second simple circuit.
- Finally, remove edge {c, g} to produce a simple graph with no simple circuits.
- This subgraph is a spanning tree, because it is a tree that contains every vertex of G.
- The sequence of edge removals used to produce the spanning tree is illustrated in Figure below.

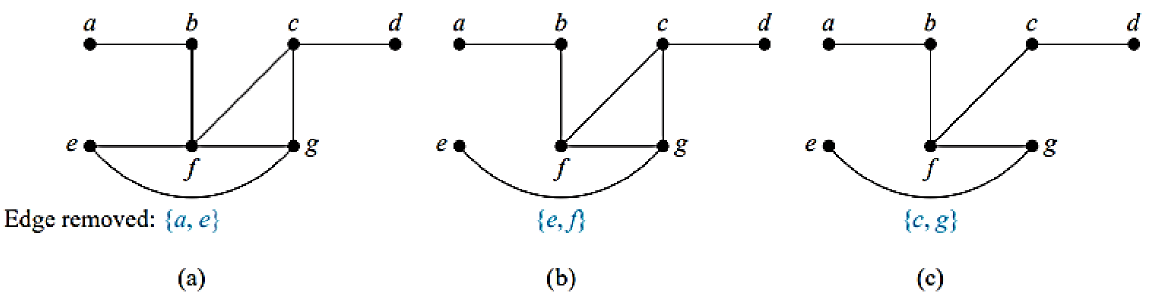


Fig: Producing a Spanning Tree for G by Removing Edges That Form Simple Circuits.

Minimum Spanning Tree

- A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.
- There are two algorithm to find the MST.
  - ➢ Prim's Algorithm
  - ➢ Kruskal's Algorithm

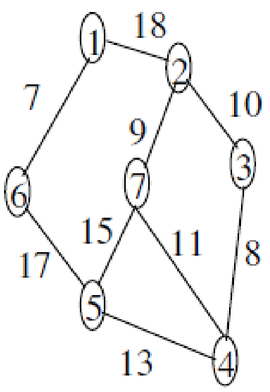
<u>Kruskal's Algorithm</u>

- To carry out Kruskal's algorithm, choose an edge in the graph with minimum weight.

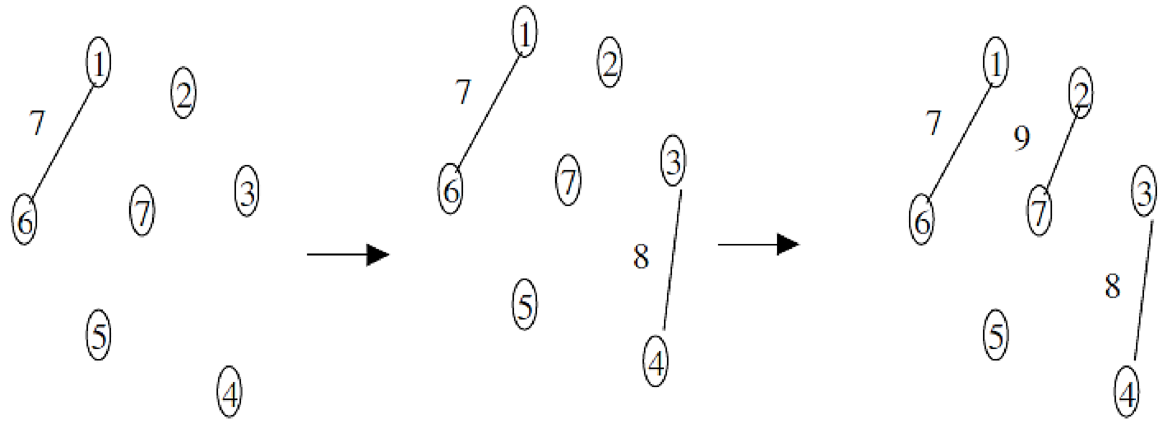
<u>Algorithm:</u>

procedure *Kruskal*(G: weighted connected undirected graph with n vertices)
T := empty graph
for $i := 1$ **to** $n - 1$
    e := any edge in G with smallest weight that does not form a simple circuit
        when added to T
    $T := T$ with e added
return T {T is a minimum spanning tree of G}

EXAMPLE: Find the MST and its weight of the graph.

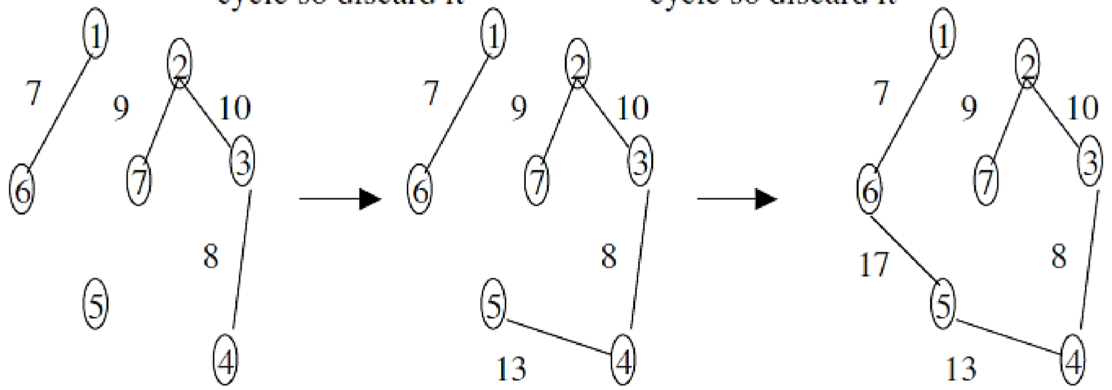


Solution:



Edge with weight 11 forms cycle so discard it

Edge with weight 15 forms cycle so discard it



- The total weight of MST is 64.

Theorem

Let G = (V, E) be a loop-free undirected graph, then G is a tree if there is a unique path between any two vertices of G.

Proof:

- Let G = (V, E) be a graph.
- Since there is a path between each pair of vertices u and v, G must be connected.
- Thus, to show G is a tree it remains to show that G has no cycles.
- Since G is loop-free, it has no cycles of length 1.
- Suppose that G has a cycle of length greater than one, say

$$C = (v_1, v_2, \ldots, v_n, v_1)$$

- Then any two distinct vertices of the cycle C are joined by two paths, which contradict the fact that there is an unique path between any two vertices of G.
- Hence, G has no cycles and so it is a tree.

Theorem

A tree with n vertices has exactly n-1 edges.

Proof:

- Let G = (V, E) be a tree with n vertices.
- We use induction method to prove it.

Basis Step:

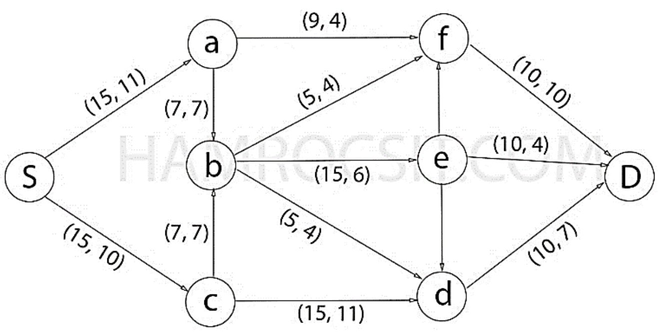
- Let n = 1 i.e. G has one vertex, since it has no loops, the number of edges in G is 0. This establishes that it is true for n = 1.

Inductive Step:

- Let it be true for n. Now we wish to show it is true for n+1.
- Let G be a tree with n+1 vertices and let u be a vertex of degree 1.
- If we remove such a vertex and the edge incident on it, then sub graph G – u is still connected and has no cycles.
- Hence G – u is a tree.
- However G – u has n vertices, so by induction it has n-1 edges.
- Since G – u has exactly one edge less than that of G, it follows that G has n edges. So assuming n+1 vertices of G we got n edges.
- This completes the proof.

Assignment

1. What is S-D cut? For the following network flow find the maximal flow from S to D. (10) [TU 2075]



2. Define Eular path and Hamilton path with examples. Draw the Hasse diagram for the divisible relation on the set { 1, 2, 5, 8, 16, 32} and find the maximal, minimal, greatest and least element if exist. (5) [TU 2075]
3. Draw the Hasse diagram for divisibility on the set {1,2,3,4,6,8,12}. Do the maximal, minimal elements exist? If so, what are they? What is the greatest element?

4. Define spanning tree and minimum spanning tree. Mention the conditions for two graphs for being isomorphic with an example. (5) [TU 2075]
5. Which of the following are possets? (5) [TU 2076]
    a. (Z, =)
    b. (Z, ≠)
    c. (Z, ⊆)
6. What is meant by chromatic number? How can you use graph colouring to schedule exams? Justify by using 10 subjects assuming that the pairs {(1,2), (1,5), (1,8), (2,4), (2,9), (2,7), (3,6), (3,7), (3,10), (4,8), (4,3), (4,10), (5,6), (5,7)} of subjects have common students. (1+4) [TU Model Question]
7. How Zero-one matrix and diagraphs can be used to represent a relation? Explain the process of identifying whether the graph is reflexive, symmetric, or anti-symmetric by using matrix or diagraph with suitable example. (4+6) [TU Model Question]
8. Define reflexive closure and symmetric closure. Find the remainder when $4x^2 - x + 3$ is divided by $x + 2$ using remainder theorem. [TU 2076]
9. Define bipartile graph with example. State the necessary conditions for the graphs to be isomorphic. [TU 2079]