GRAPH THEORY
DISCRETE STRUCTURE

Sudip Rana

Motivation

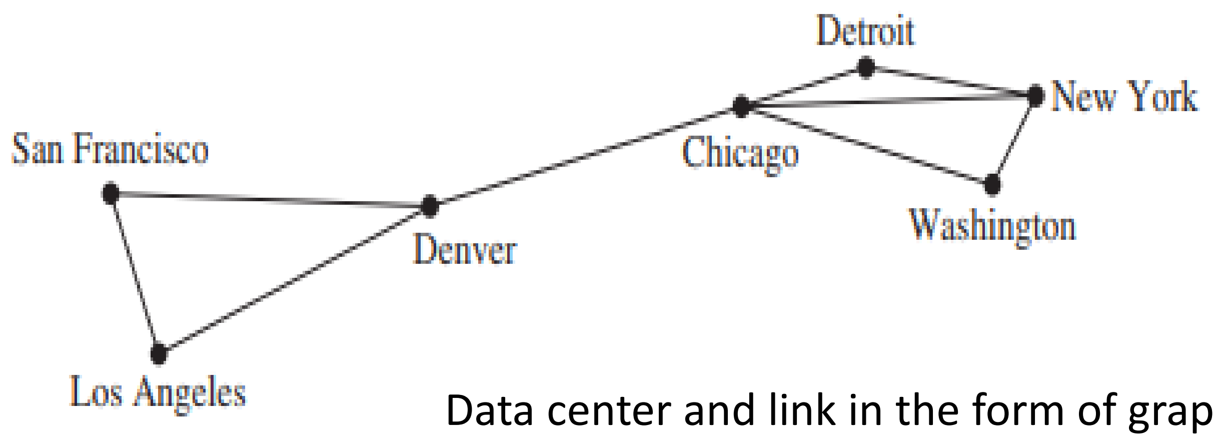
- Graphs are used as models in a variety of areas.
- Graphs are used to represent the competition of different species in an ecological niche
- Graphs are used to represent who influences whom in an organization.
- Graphs are used to represent the outcomes of round-robin tournaments.
- Graphs can be used to model acquaintanceships between people, collaboration between researchers, telephone calls between telephone numbers, and links between websites.

Contd….

- graphs can be used to model roadmaps and the assignment of jobs to employees of an organization.
- Using graph models, we can determine whether it is possible to walk down all the streets in a city without going down a street twice, and we can find the number of colors needed to color the regions of a map.
- Graphs with weights assigned to their edges can be used to solve problems such as finding the shortest path between two cities in a transportation network etc…..

Introduction

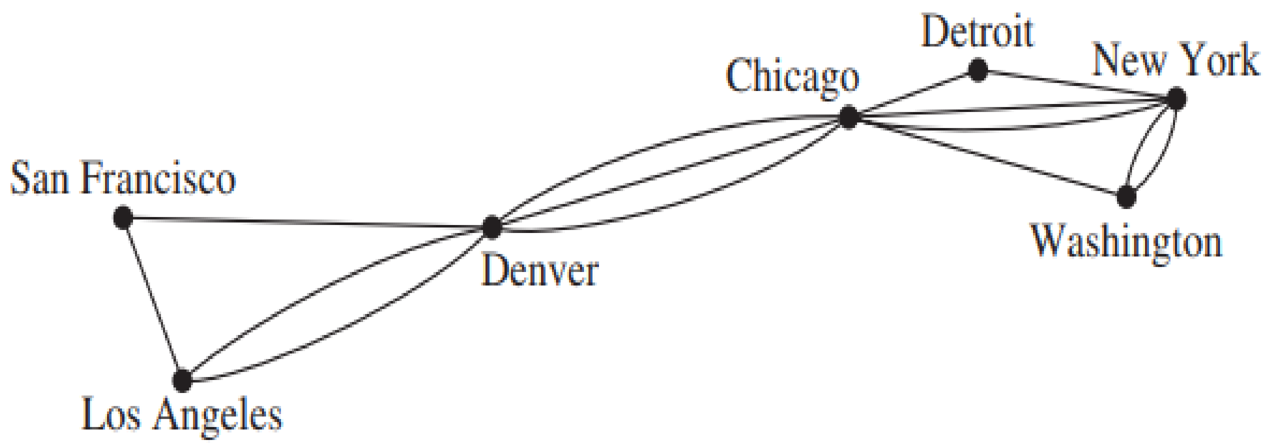
- A graph G = (V , E) consists of V , a nonempty set of vertices (or nodes) and E, a set of edges.
- Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.



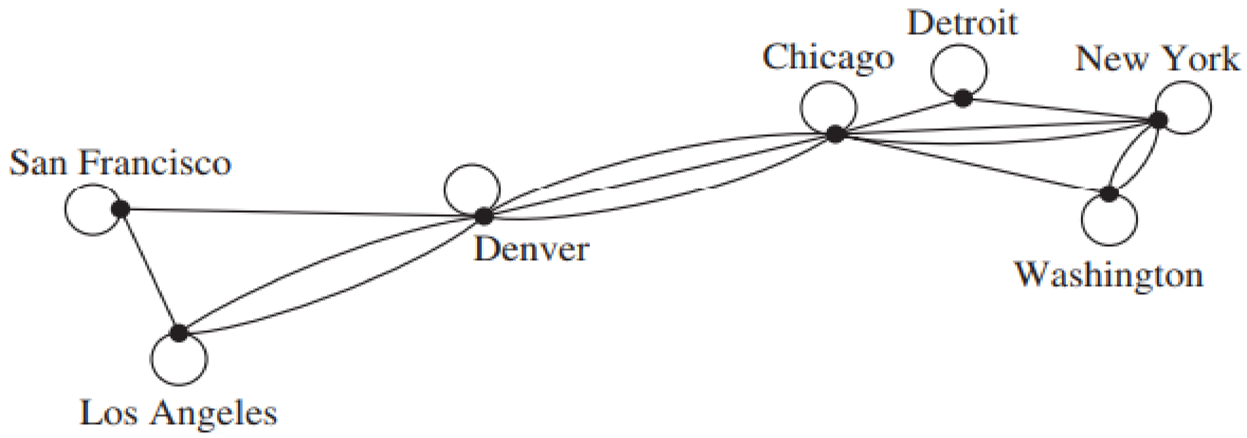
Data center and link in the form of graph

Contd…

- 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.
- Graphs that may have **multiple edges** connecting the **same vertices** are called multigraphs.
- Graphs that may include **loops**, and possibly **multiple edges** connecting the same pair of vertices or a vertex to itself, are sometimes called pseudo graphs.



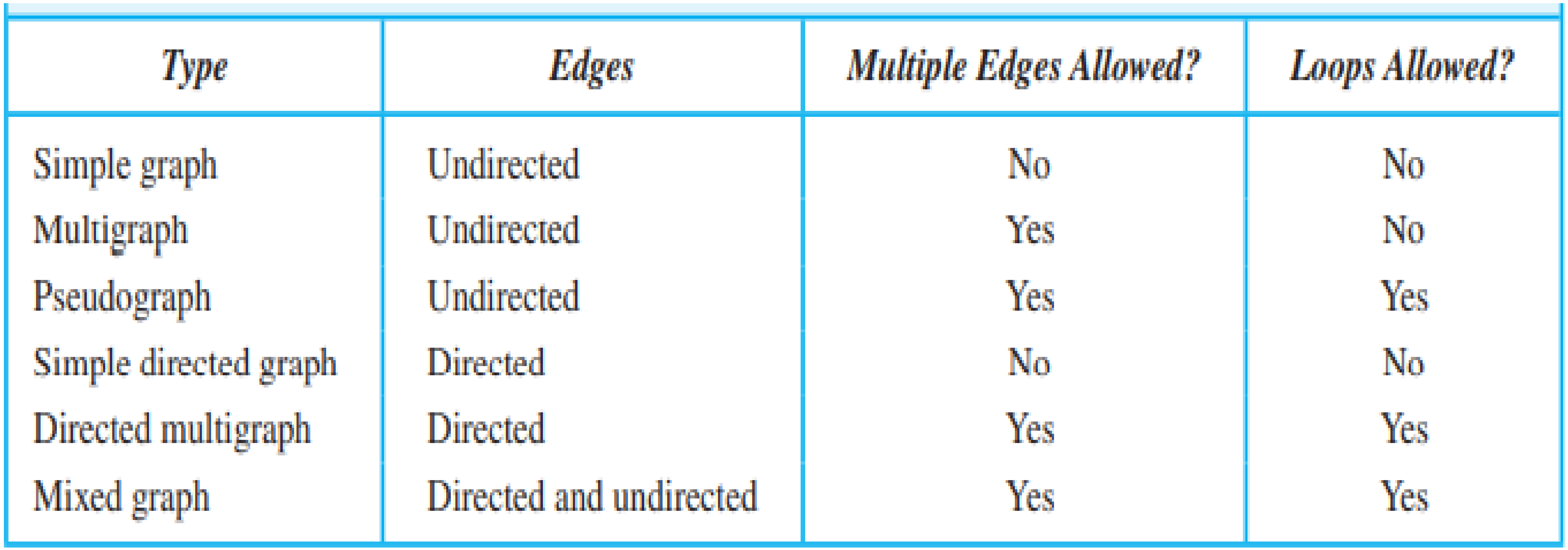
Multiple edge connecting vertices



Pseudo graph

Directed graph

- Aforementioned graph are undirected graph. But it may be necessary to give the direction to the edge.
- 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.
- So we can have simple directed graph, directed multigraphs, mixed graph.



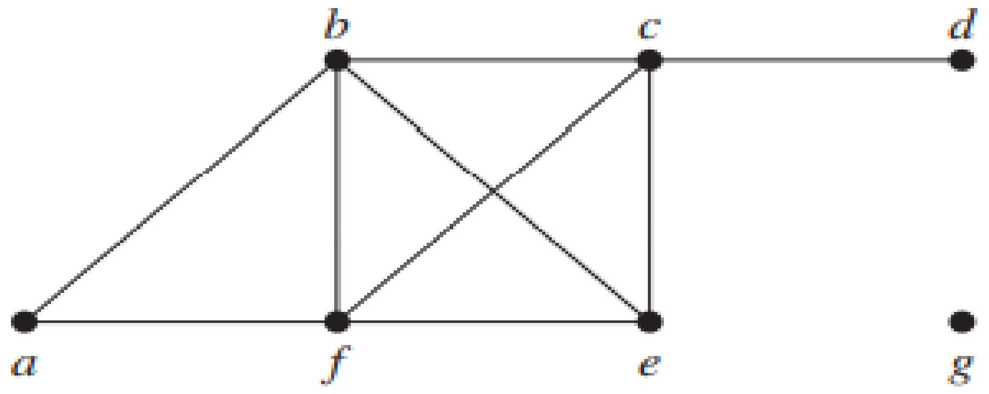
Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Graph terminology

Terminology of graph

- Two vertices u and v in an undirected graph G are called adjacent (or neighbors) 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.
- The set of all neighbors of a vertex v of G = (V , E), denoted by N (v), is called the neighborhood of v.
- 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…



- deg(a) = 2, deg(b) = deg(c) = deg(f) = 4, deg(d) = 1, deg(e) = 3, and deg(g) = 0. The neighborhoods 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) = ∅.

Contd…

- A vertex of degree zero is called isolated. It follows that an isolated vertex is not adjacent to any vertex.
- Vertex g in graph 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 above is pendant.

Handshaking theorem

- Let G = (V , E) be an undirected graph with m edges.
- Then according to this theorem $\sum_{v \in V} \deg(v) = 2m$
- (Note that this applies even if multiple edges and loops are present)
- It is also called as sum of degree of vertices.
- How many edges are there in a graph with 10 vertices each of degree six?
- Solution: Because the sum of the degrees of the vertices is 6 · 10 = 60, it follows that 2m = 60 where m is the number of edges. Therefore, m = 30.

An undirected graph has an even number of vertices of odd degree.

- Let V1 and V2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph G = (V , E) with m edges. Then

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

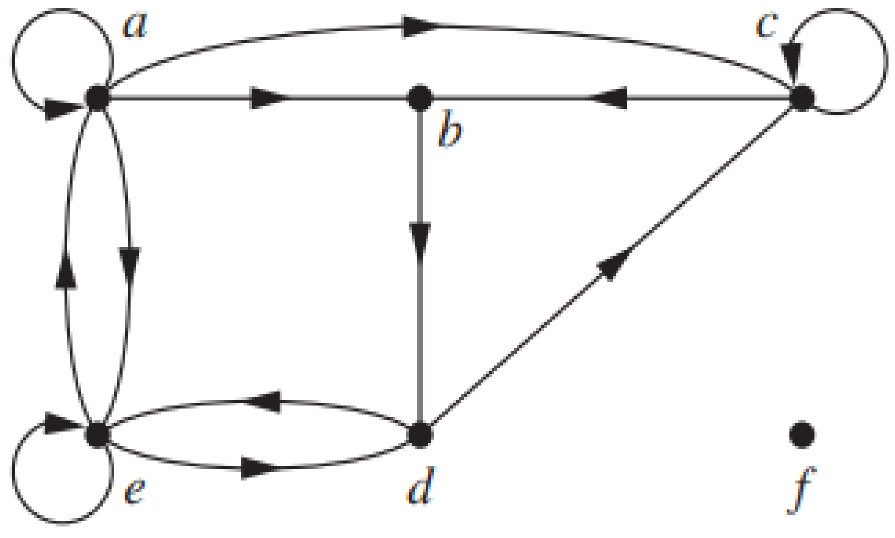
- Here the first term is even. Second term is odd.
- But overall the summation is equal to even. Therefore there must be even number of vertices of odd degree.

Contd…

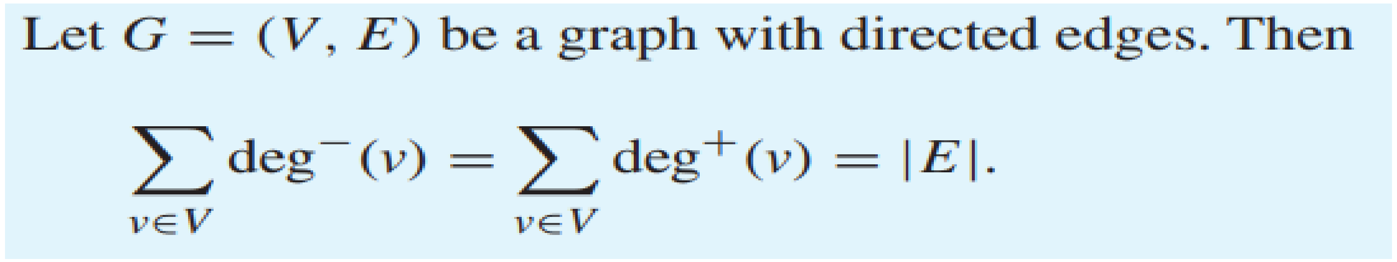
- When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v and v is said to be adjacent from u.
- The vertex u is called the initial vertex of (u, v), and v is called the terminal or end vertex of (u, v).
- The initial vertex and terminal vertex of a loop are the same.
- 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.

Find the in-degree and out-degree of each vertex in the graph G with directed edges

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.



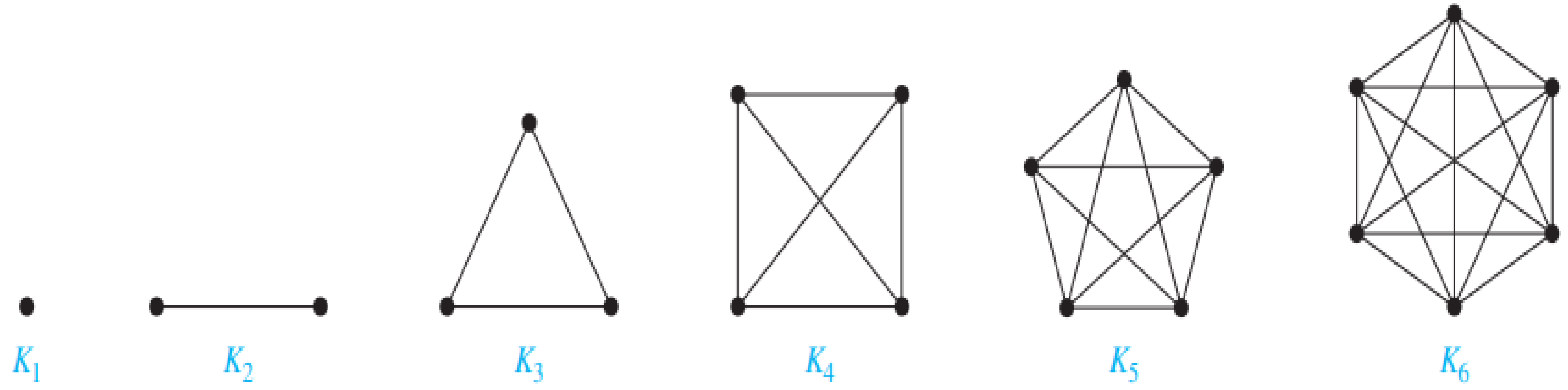
G


Let $G = (V, E)$ be a graph with directed edges. Then

$$\sum_{v\in V} \deg^-(v) = \sum_{v\in V} \deg^+(v) = |E|.$$

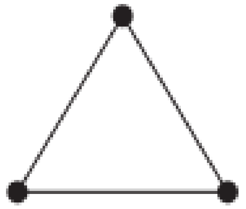
Special simple graph

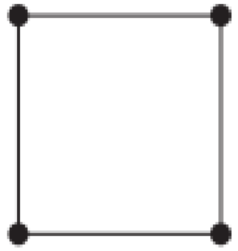
- Complete Graphs: A complete graph on n vertices, denoted by k_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.

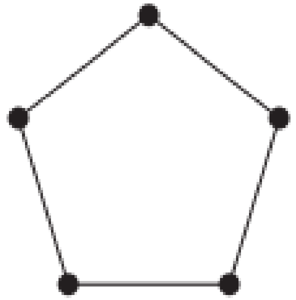


Contd…

- Cycles: A cycle C_n, n ≥ 3, consists of n vertices v1, v2,..., vn and edges {v1, v2}, {v2, v3},...,{vn−1, vn}, and {vn, v1}.
- Wheels: We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for n ≥ 3, and connect this new vertex to each of the n vertices in C_n, by new edges.
- 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.

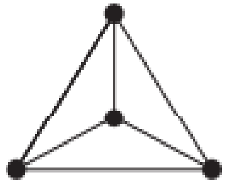

C_3

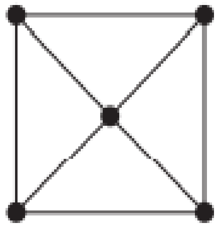

C_4

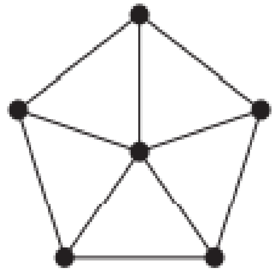

C_5

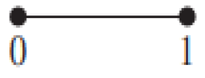
Cycle graph

Wheel graph

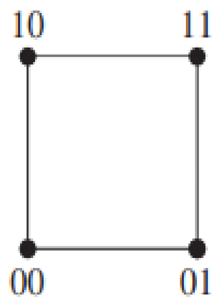

W_3


W_4

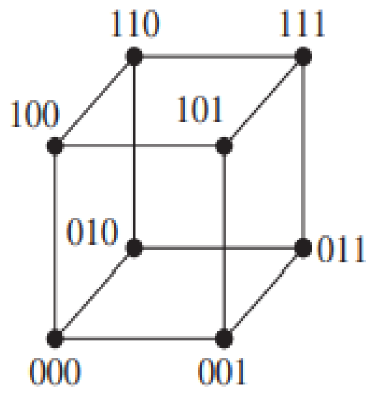

W_5



Q_1



Q_2



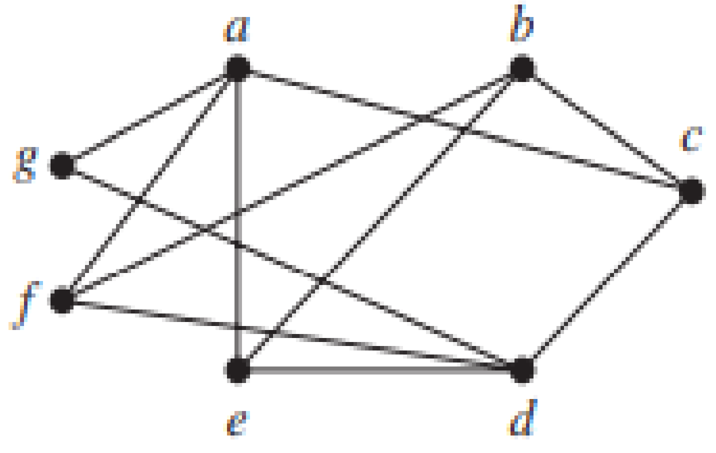
Q_3

N cube

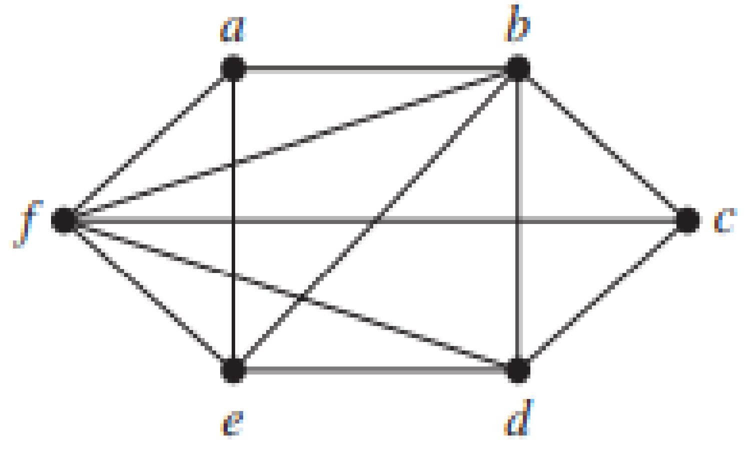
Bipartite graph

- A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V1 and V2 such that every edge in the graph connects a vertex in V1 and a vertex in V2.
- No edge in G connects either two vertices in V1 or two vertices in V2.
- When this condition holds, we call the pair (V1, V2) a bipartition of the vertex set V of G.

Are following graph bipartite?



G



H

Contd…

- 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. (Note that for G to be bipartite it is not necessary that every vertex in {a, b, d} be adjacent to every vertex in {c, e, f, g}. For instance, b and g are not adjacent.)
- Graph H is not bipartite because its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset. (The reader should verify this by considering the vertices a, b, and f .)

Assignment…

- There are some example of bipartite graph application in Kenneth rosen book page 658. go through it

Graph isomorphism

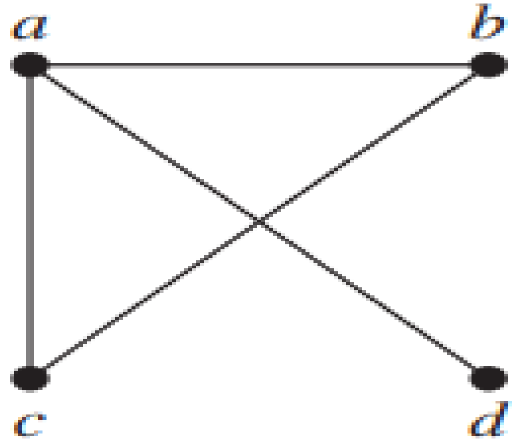
- One way to obtain information from graph is to represent it in the form of matrices.
- Suppose that G = (V , E) is a simple graph where |V | = n. Suppose that the vertices of G are listed arbitrarily as v1, v2,..., vn.
- The adjacency matrix A (or A_G) of G, with respect to this listing of the vertices, is the n x n zero–one matrix with 1 as its (i, j)th entry when vi and vj are adjacent, and 0 as its (i, j)th entry when they are not adjacent.

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

Use an adjacency matrix to represent the graph

- We order the vertices as a, b, c, d. The matrix representing this graph is

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



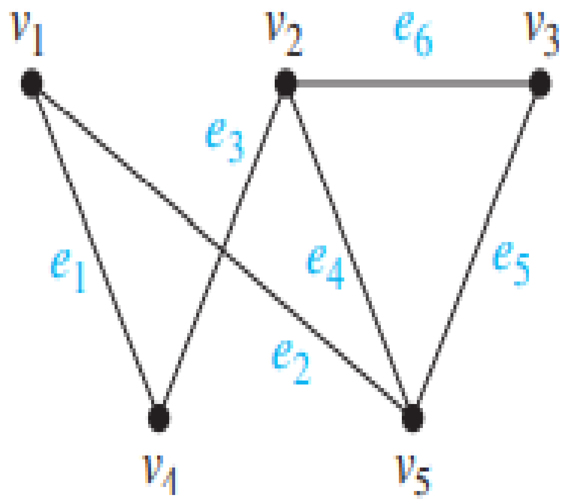
there may be as many as n! different adjacency matrices for a graph with n vertices, because there are n! different orderings of n vertices.

Contd…

- Another common way to represent graphs is to use incidence matrices.
- Let G = (V , E) be an undirected graph. Suppose that v1, v2,..., vn are the vertices and e1, e2,...,em are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the n × m matrix M = [mij], where

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

Represent the graph shown in Figure with an incidence matrix.



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

Isomorphism

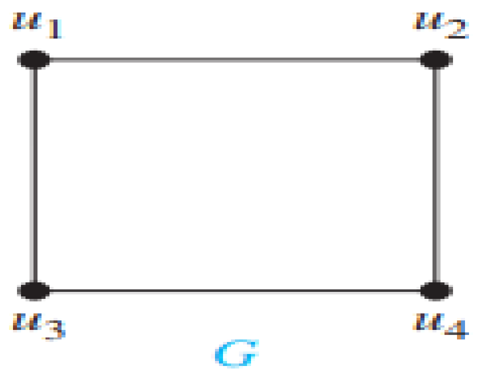
- We often need to know whether it is possible to draw two graphs in the same way.
- Do the graphs have the same structure when we ignore the identities of their vertices?
- The simple graphs G1 = (V1, E1) and G2 = (V2, E2) are isomorphic if there exists a one-to-one and onto function f from V1 to V2 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.

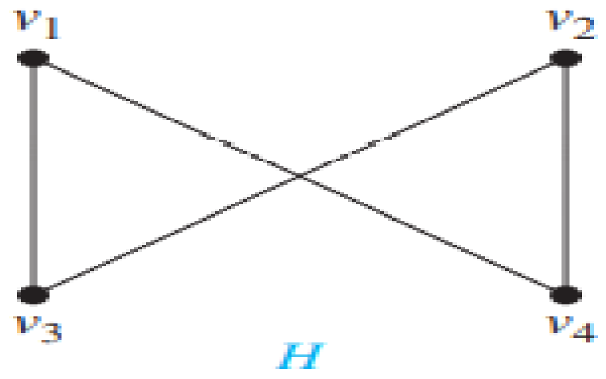
Contd….

- In other words, when two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that **preserves the adjacency relationship**.
- They also have exact structural properties(number of vertices, edges, degree of vertices.)
- Show that the graphs G = (V , E) and H = (W, F), displayed in Figure are isomorphic.

Solution…

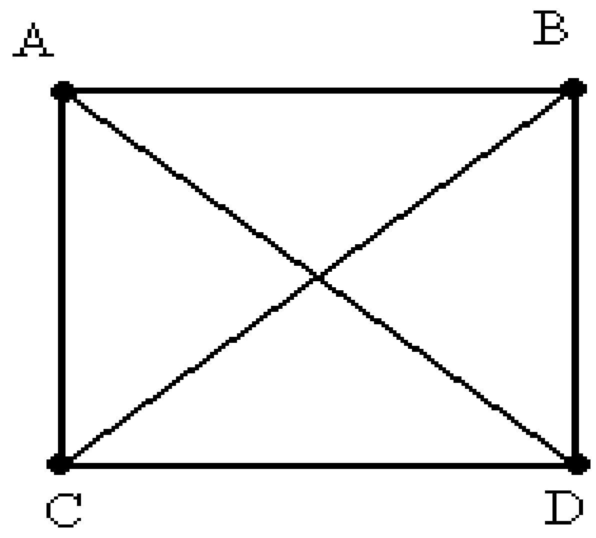
- Here both graph have same number of Vertices, edges and number of degree.
- Now, The function f with f (u1) = v1, f (u2) = v4, f (u3) = v3, and f (u4) = v2 is a one-to-one correspondence between V and W. To see that this correspondence preserves adjacency, note that adjacent vertices in G are u1 and u2, u1 and u3, u2 and u4, and u3 and u4, and each of the pairs f (u1) = v1 and f (u2) = v4, f (u1) = v1 and f (u3) = v3, f (u2) = v4 and f (u4) = v2, and f (u3) = v3 and f (u4) = v2 consists of two adjacent vertices in H.



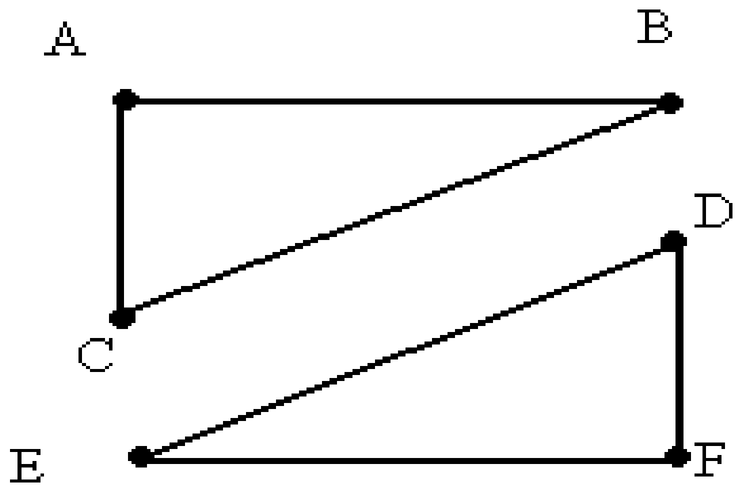


Connectivity

- **Path**: path is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph.
- **Circuit**: 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.
- Note that edges shouldn’t repeat.
- **Connected graph**: a graph is said to be connected when there is path between every pair of vertices.



Connected graph

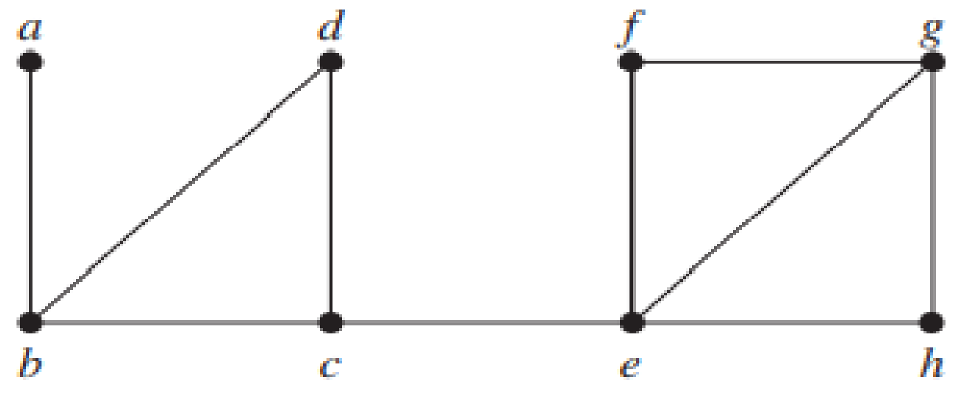


Disconnected graph

Cut vertices and cut edges

- Cut vertex: the vertex whose removal from the graph produce a sub graph such that they are disconnected.
- Cut edge: those edge whose removal from the graph produce a sub graph such that they are disconnected.
- Connected graphs without cut vertices and cut edges are called non-separable graphs.

Find the cut vertex and cut edge in graph shown below



G_1

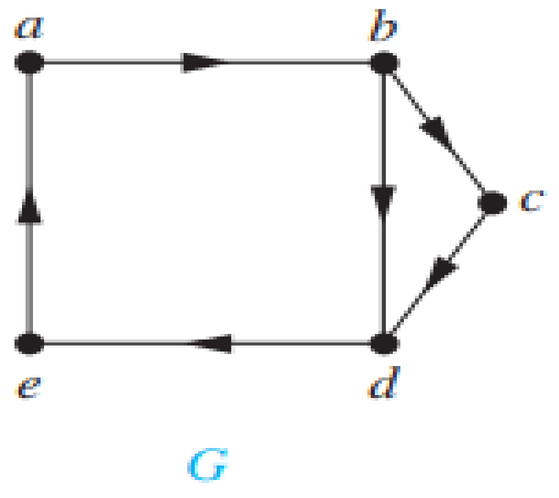
- The cut vertices of G1 are b, c, and e. The removal of one of these vertices (and its adjacent edges) disconnects the graph.
- The cut edges are {a, b} and {c, e}. Removing either one of these edges disconnects G1.

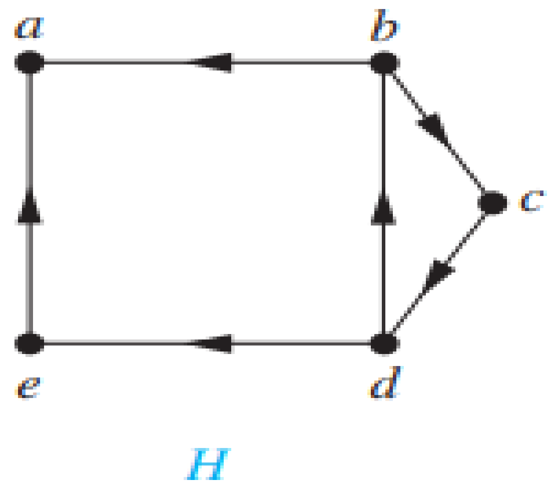
Connectedness in directed graph

- Strongly connected: 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.
- Weakly connected: A directed graph is weakly connected if there is a path between every two vertices in the underlying undirected graph.
- Any strongly connected directed graph is also weakly connected.

Are the directed graphs G and H shown in Figure strongly connected? Are they weakly connected?

- G is **strongly connected** because there is a path between any two vertices in this directed graph (the reader should verify this). Hence, G is also weakly connected. The graph **H is not strongly connected**. There is no directed path from a to b in this graph. However, H is weakly connected, because there is a path between any two vertices in the underlying undirected graph of H (the reader should verify this).

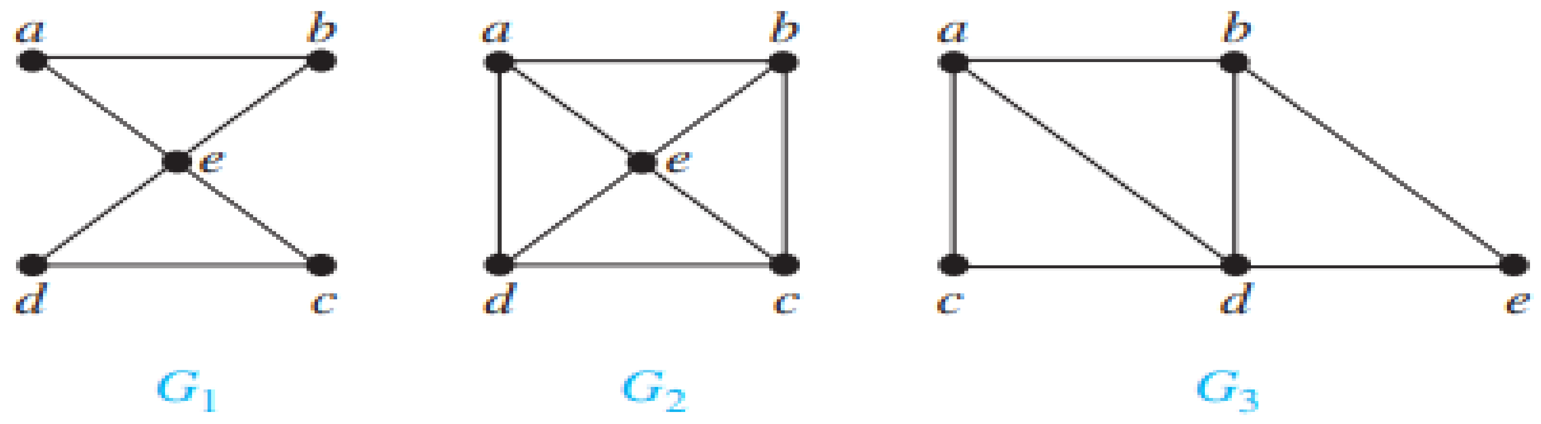




Euler graph

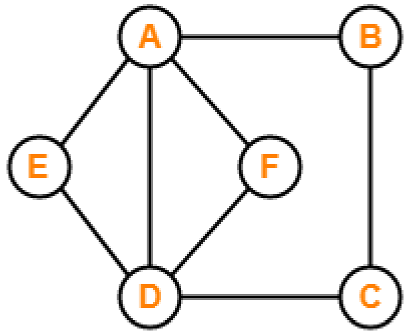
- Can we travel along the edges of a graph starting at a vertex and returning to it(same vertex) by traversing each edge of the graph exactly once?
- An Euler circuit in a graph G is a simple circuit containing every edge of G.
- An Euler path in G is a simple path containing every edge of G.
- A graph is said to be Euler graph if it contain Euler circuit.

Which of the undirected graphs in Figure below have an Euler circuit? Of those that do not, which have an Euler path?

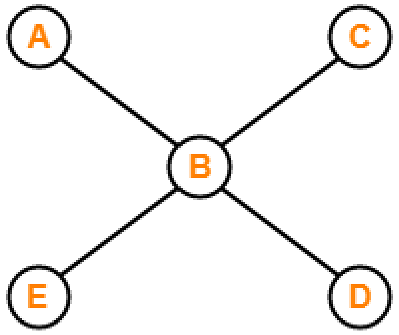


- Solution: The graph G1 has an Euler circuit, for example, a, e, c, d, e, b, a. Neither of the graphs G2 or G3 has an Euler circuit.
- However, G3 has an Euler path, namely, a, c, d, e, b, d, a, b. G2 does not have an Euler path.

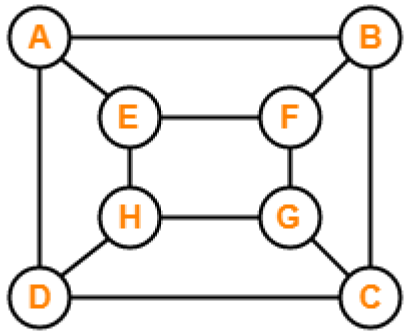
A)



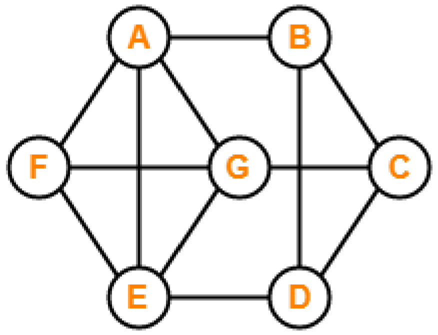
B)



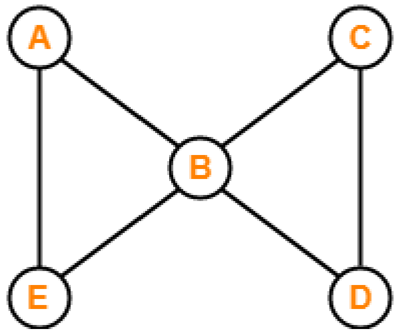
C)



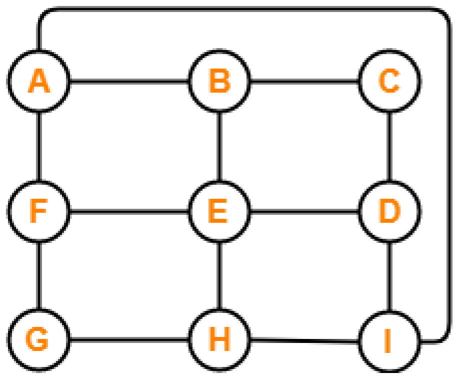
D)



E)



F)



Necessary condition to be Euler graph

- For any starting vertices, it must have at least 2 edge to start and end at same vertex.
- Similarly for any intermediate vertex, if we enter at that vertex than as it is not ending vertex we must come out also.
- Therefore what we can conclude is that if the degree of vertices in the graph is even then it is Euler graph.
    1. Graph must be connected.
    2. The degree of the vertices must be even
    3. The in degree and out degree should be equal for directed graph.

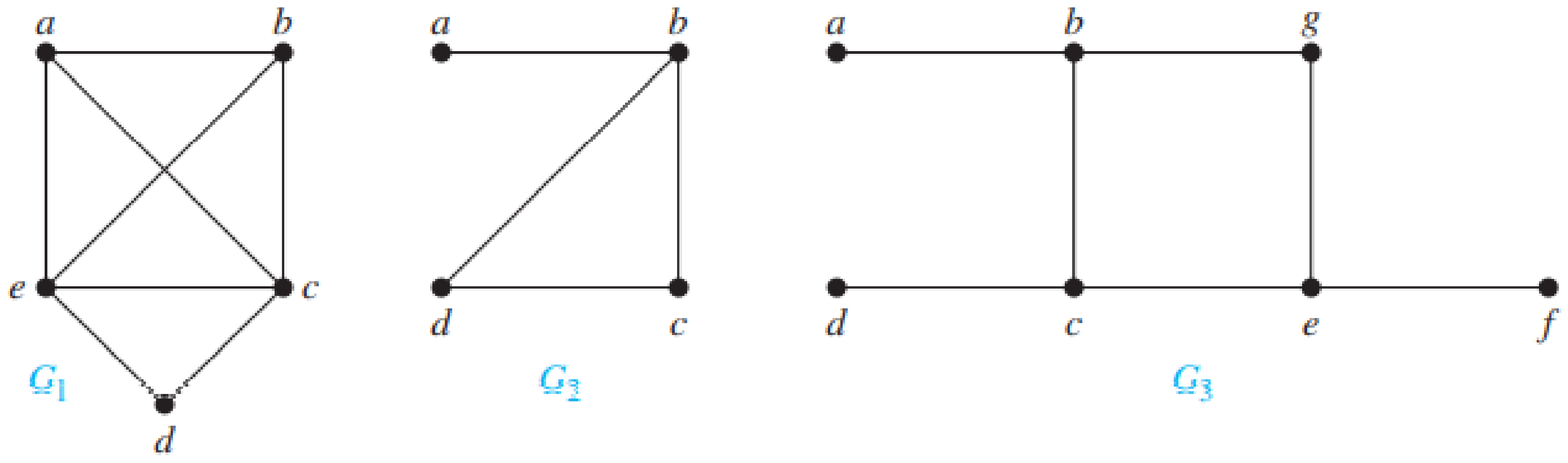
solution

A) It is an Euler graph.
B) It is not an Euler graph.
C) It is not an Euler graph.
D) It is not an Euler graph.
E) It is an Euler graph.
F) It is not an Euler graph.

Hamilton graph

- Can we have simple paths and circuits that contain **every vertex of the graph exactly once**?
- A simple path in a graph G that passes through **every vertex exactly once** is called a **Hamilton path**.
- A simple circuit in a graph G that passes through **every vertex exactly once** is called a **Hamilton circuit** except starting and ending vertex which is same.
- It can be **thought like one more restriction to Euler graph**. While traversing the vertex, if any edge can also be omitted.

Which of the simple graphs in Figure below have a Hamilton circuit or, if not, a Hamilton path?



- Solution: G1 has a Hamilton circuit: a, b, c, d, e, a. There is no Hamilton circuit in G2 (this can be seen by noting that any circuit containing every vertex must contain the edge {a, b} twice), but G2 does have a Hamilton path, namely, a, b, c, d. G3 has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges {a, b}, {e, f }, and {c, d} more than once.

Contd…

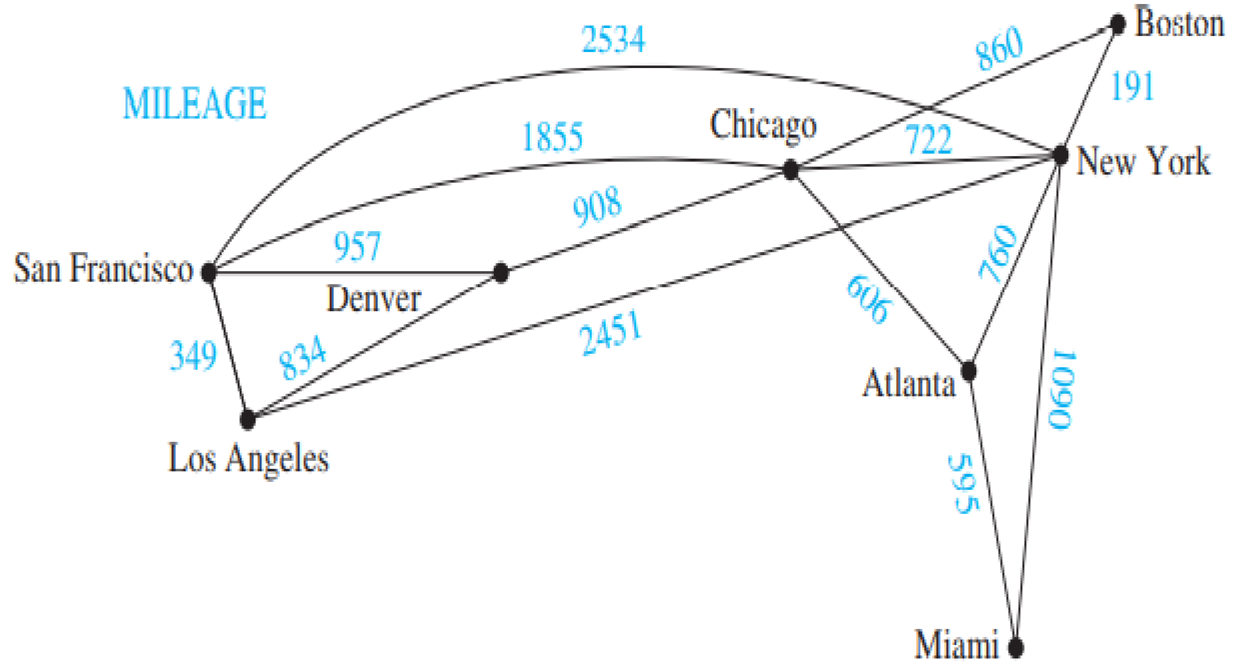
- Note:
- In Hamiltonian path, **all the edges may or may not be covered but edges must not repeat.**
- Any Hamiltonian circuit can be converted to a Hamiltonian path by removing one of its edges.
- Every graph that contains a Hamiltonian circuit also contains a Hamiltonian path but vice versa is not true.
- **There may exist more than one Hamiltonian paths and Hamiltonian circuits in a graph.**
- And there is also no any condition or rules like Euler to know whether it is Hamilton or not?

solution

- Not
- Not
- Yes
- Yes
- Not
- Yes

Shortest path problem

- Many problems can be modeled using graphs with **weights** assigned to their edges.
- As an illustration, consider how an airline system can be modeled. We set up the basic graph model by representing cities by vertices and flights by edges.
- Problems involving distances (weight) can be modeled by assigning distances between cities to the edges.



Contd …

- 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.
- Determining a path of **least length** between two vertices in a network is very important task.

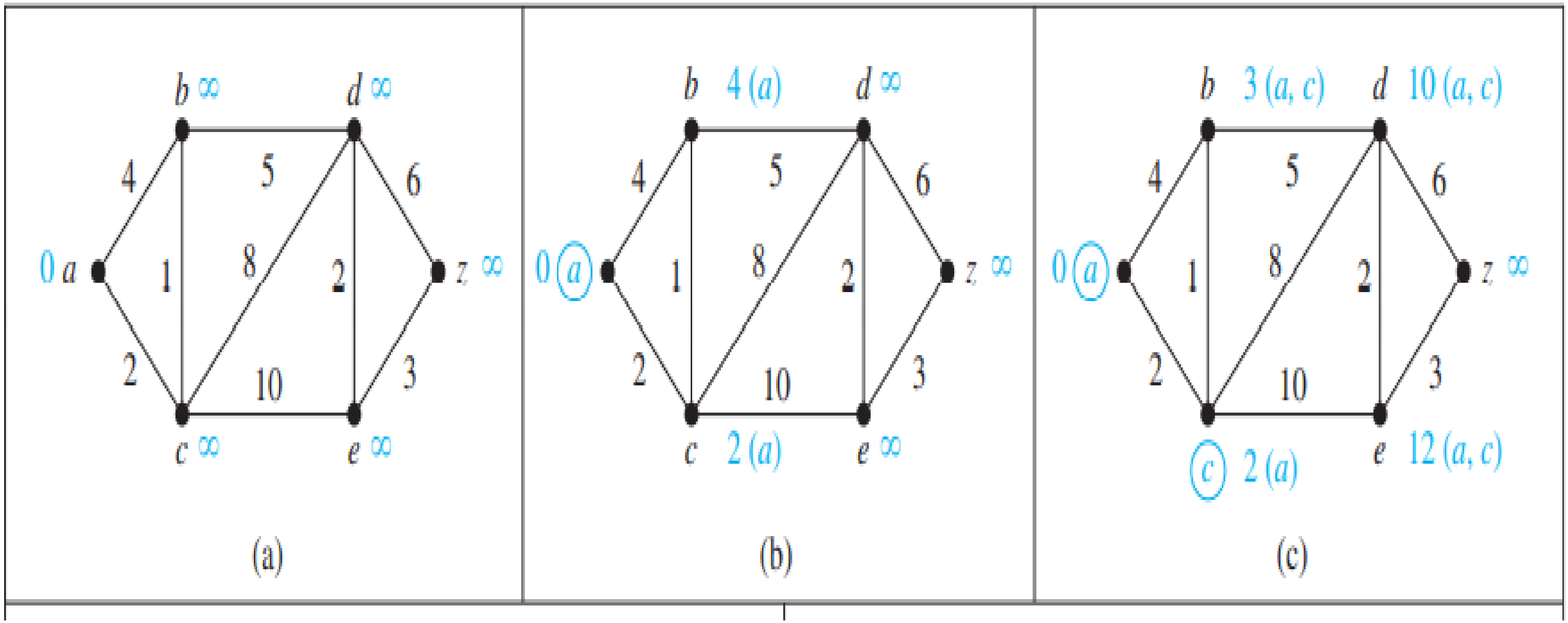
Shortest path algorithm

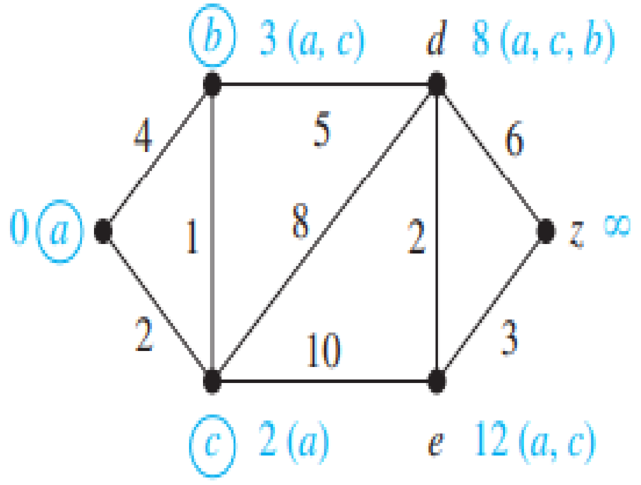
- There are several different algorithms that find a shortest path between two vertices in a weighted graph.
- Dijkstra’s algorithm is one of the popular algorithm to find shortest path(cost) in this problem
- It is based on the relaxation of vertices after finding the cost.
- If d[u]+ c(u, v) < d[v]

    then d[v]= d[u]+ c(u, v)

Where (u, v) are the vertices, d[u] is the cost up to vertex ‘u’ and c(u, v) is the cost from vertex u to v.

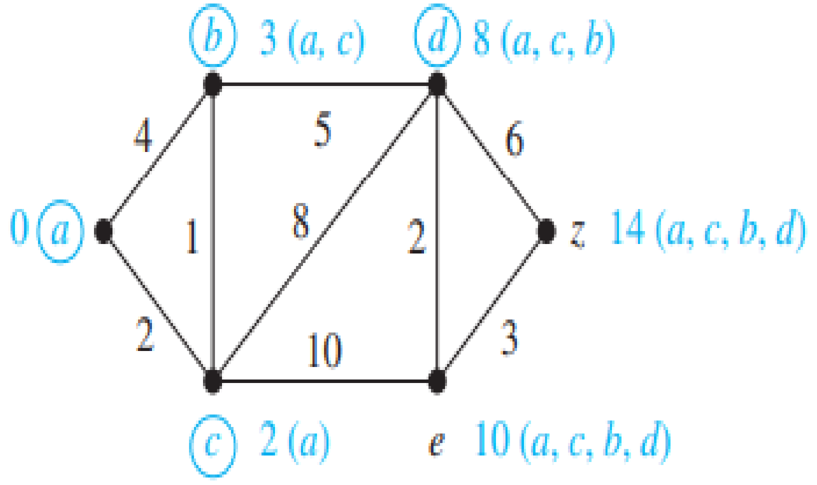
Use Dijkstra's algorithm to find the length of a shortest path between the vertices a and z in the weighted graph(graphically)





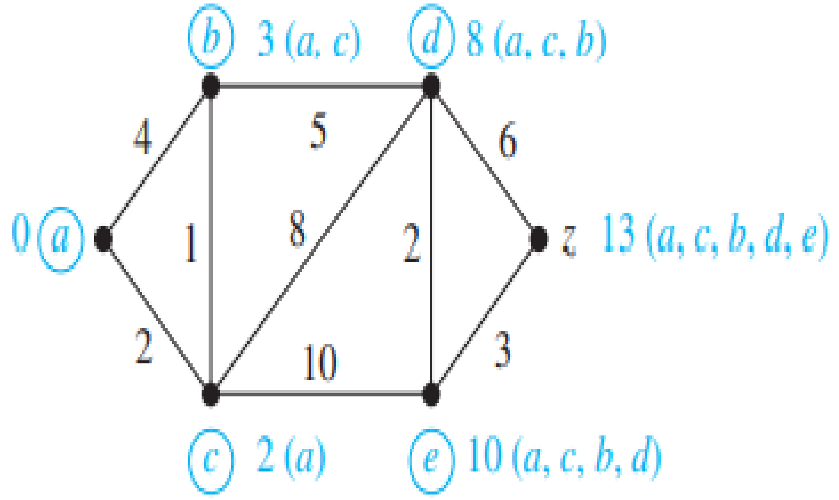
b 3 (a, c)
d 8 (a, c, b)
4
5
6
0 a
1
8
2
z ∞
2
10
3
c 2 (a)
e 12 (a, c)

(d)



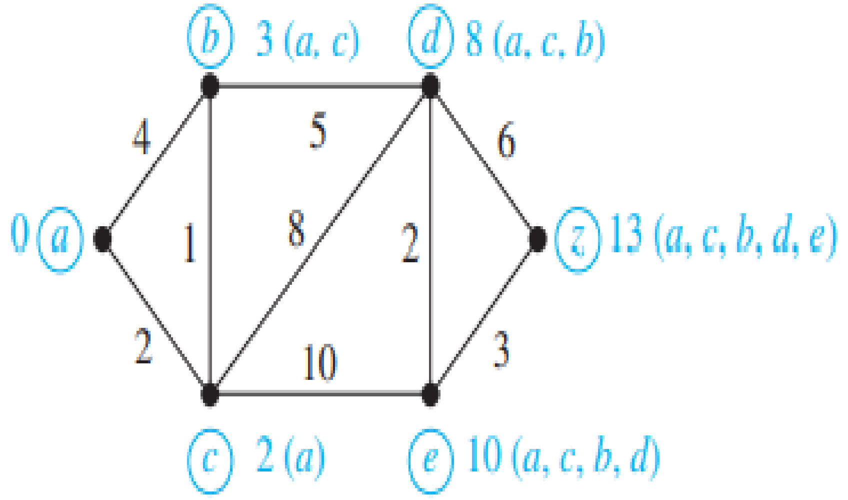
b 3 (a, c)
d 8 (a, c, b)
4
5
6
0 a
1
8
2
z 14 (a, c, b, d)
2
10
3
c 2 (a)
e 10 (a, c, b, d)

(e)



b 3 (a, c)
d 8 (a, c, b)
4
5
6
0 a
1
8
2
z 13 (a, c, b, d, e)
2
10
3
c 2 (a)
e 10 (a, c, b, d)

(f)



b 3 (a, c)
d 8 (a, c, b)
4
5
6
0 a
1
8
2
z 13 (a, c, b, d, e)
2
10
3
c 2 (a)
e 10 (a, c, b, d)

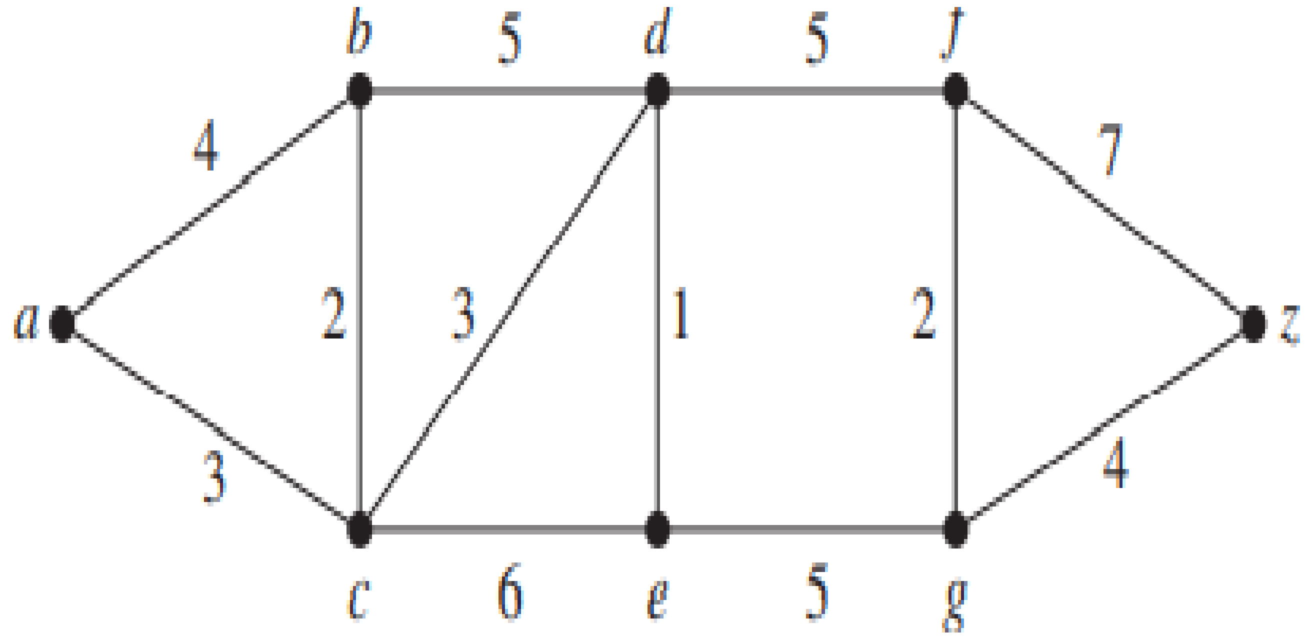
(g)

Table method

Selected vertex	b	C	d	e	z
c	4	2	∞	∞	∞
b	3	-	10	12	∞
d	-	-	8	12	∞
e	-	-	-	10	14
z	-	-	-	-	13

The path from a to z is a-c-b-d-e-z and total cost =13

Find the cost metric from vertex a to each vertex and also length of shortest path from a to z



Planer graph

- Can a graph be drawn in the plane without its edges crossing?
- A graph is called planar if it can be drawn in the plane without any edges crossing.
- Such a drawing is called a planar representation of the graph
- A graph **may be planar** even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.
- Consider the example shown in next slide.

Example…

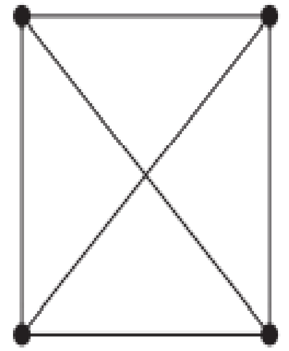


FIGURE 2 The Graph K_4.

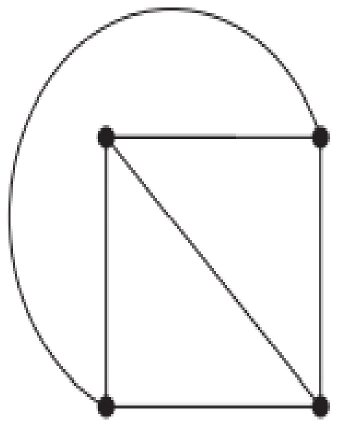


FIGURE 3 K_4 Drawn with No Crossings.

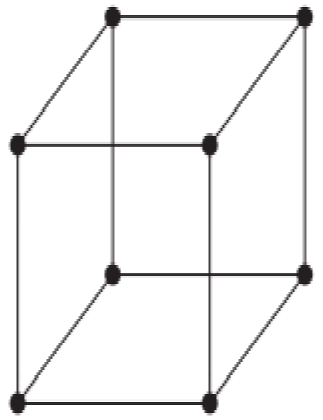


FIGURE 4 The Graph Q_3.

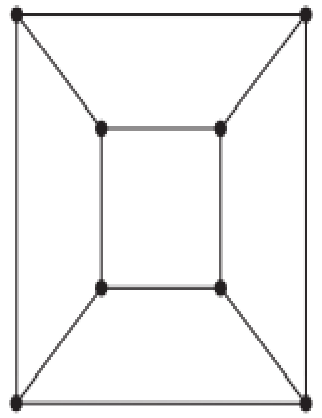


FIGURE 5 A Planar Representation of Q_3.

Try for k_5 and $k_{3,3}$

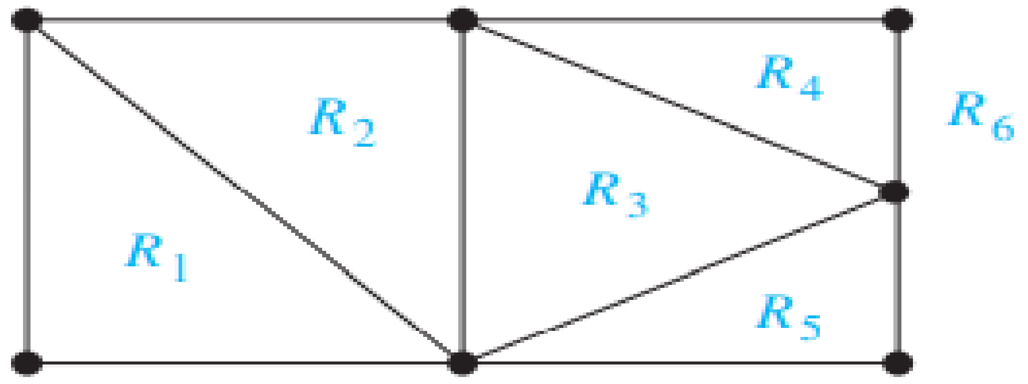
- Both are the simplest non planar graph in terms of vertices and edges.

Application of planar graph

- Planarity of graphs plays an important role in the design of electronic circuits.
- We can print a **circuit on a single board** with no connections crossing if the graph representing the circuit is planar.
- The planarity of graphs is also useful in the **design of road networks**. Suppose we want to connect a group of cities by roads. We can model a road network connecting these cities using a simple graph with vertices representing the cities and edges representing the highways connecting them.
- We can built this road network without using underpasses or overpasses if the resulting graph is planar.

Euler's formula

- Let G be a connected **planar simple graph** with ‘e’ edges and ‘v’ vertices. Let “r” be the number of regions in a planar representation of G.
- Then r = e − v + 2.
- Region can be either bounded or unbounded.



Derivation from Euler formula

- For any connected planar graph the inequality 2e$\geq 3r$ always holds true.
- **Corollary 1:** If G is a connected planar simple graph with e edges and v vertices, where v ≥ 3,
- Then **e ≤ 3v – 6** (solving Euler formula and above inequality).
- Using this formula we can show k_5 is non planar.
- **Corollary 2:** If a connected planar simple graph has e edges and v vertices with v ≥ 3 and no circuits of length three, then e ≤ 2v − 4.
- Using this we can show K3,3 is nonplanar.

Regular graph

- A graph is said to be regular if the degree of all vertices in the graph is equal.
- It is called k regular graph.
- For eg. k_4 is called 3-regular graph.
- Complete graph is also regular but not vice versa.

Graph coloring

- Consider two map shown in the figure.
- When a map is colored, **two regions with a common border** are customarily assigned different colors.
- One way to ensure that two adjacent regions never have the same color is to use a different color for each region.
- The map shown on the left in Figure 1, four colors suffice, but three colors are not enough. In the map on the right in Figure 1, three colors are sufficient (but two are not).

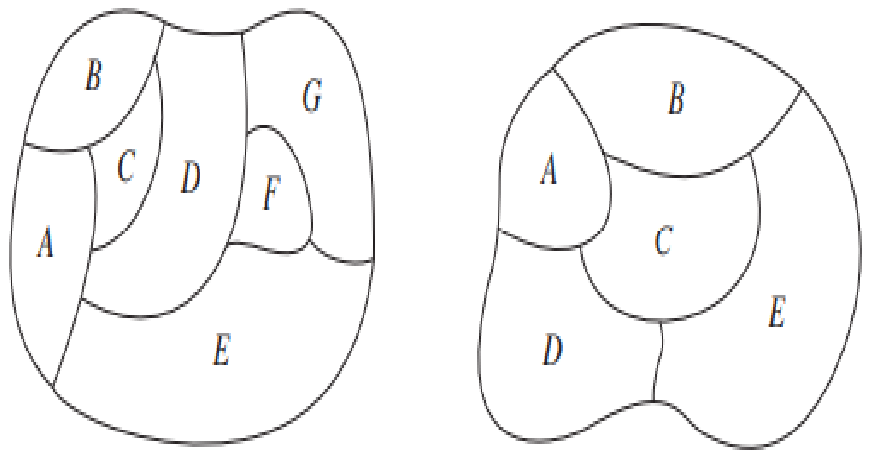
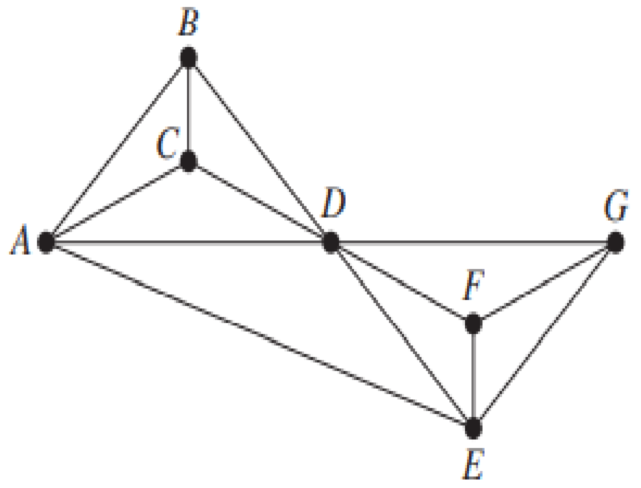
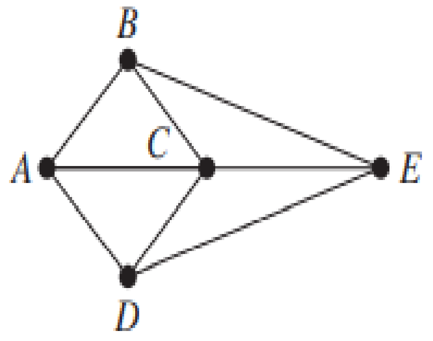


FIGURE 1 Two Maps.

Contd…

- We can convert the problem into dual graph.
- Each region of the map is represented by a vertex.
- Edges connect two vertices if the regions represented by these vertices have a common border.





Contd…

- The problem of coloring the regions of a map is equivalent to the problem of coloring the vertices of the dual graph so that **no two adjacent vertices in this graph have the same color**.
- A graph can be colored by assigning a different color to each of its vertices.
- However, for most graphs, a coloring can be found that uses fewer colors than the number of vertices in the graph.
- What is the least number of colors necessary?

Chromatic number

- The **chromatic number** of a graph is the least number of colors needed for a coloring of this graph.
- The chromatic number of a graph G is denoted by χ (G). (Here χ is the Greek letter chi.)
- What is the chromatic number of k_n?

Ans: every vertices are connected to each other incase of complete simple graph. Hence, the chromatic number of Kn is n.

That is, χ (Kn) = n.

Contd…

- What is the chromatic number of the complete bipartite graph $k_{m,n}$, where m and n are positive integers?

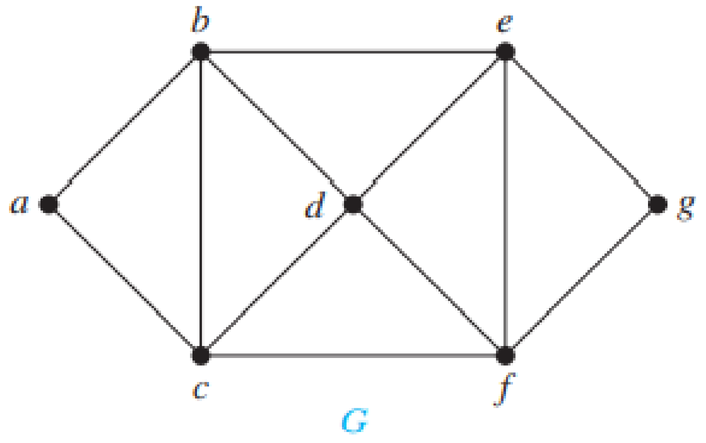
Ans: since it is bipartite graph, we can ,construct two sets disjoint with each other in same set. Hence, χ (Km,n) = 2.

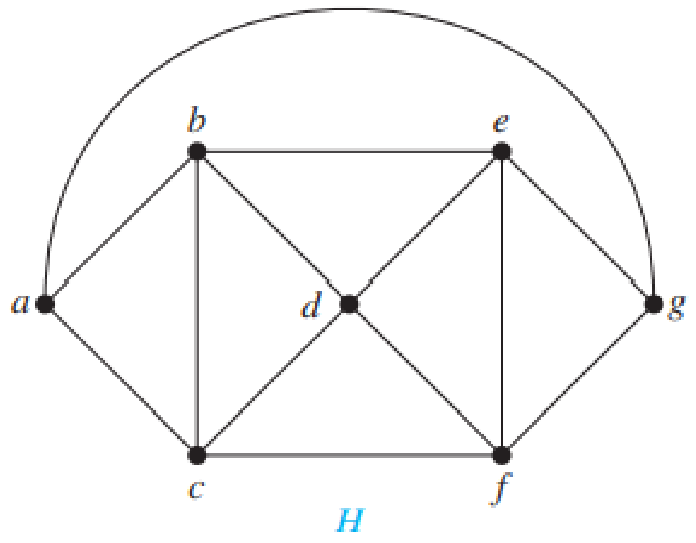
- What is the chromatic number of the graph Cn, where n ≥ 3?

Ans: chromatic number for the cycle graph with even vertices is 4 and for odd vertices is 3. (verify yourself)

Four color Theorem

- The chromatic number of a **planar graph** is **no greater than four**.
- Note that the four color theorem **applies only to planar graphs**. Nonplanar graphs can have arbitrarily large chromatic numbers.





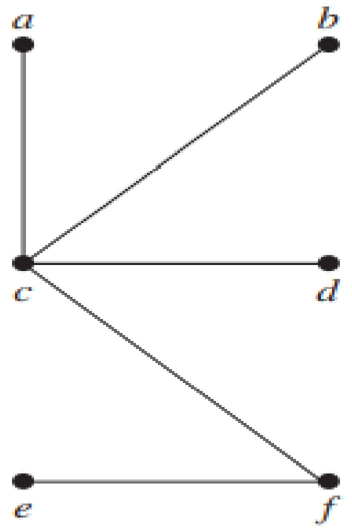
Assignment

- Scheduling Final Exams
- How can the final exams at a university be scheduled so that no student has two exams at the same time?
- Please go through page number 731 to find the answer.

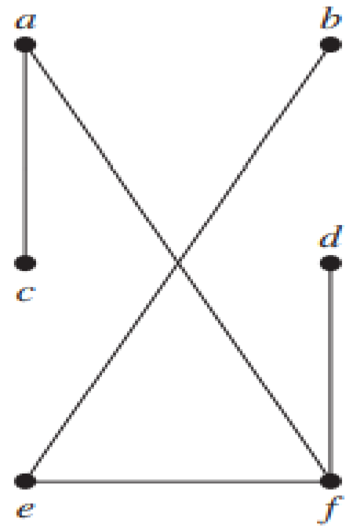
Tree

- A **connected graph** that contains no **simple circuits(cycle)** is called a tree.
- Because a tree cannot have a simple circuit, a **tree cannot contain multiple edges or loops.**
- Therefore any tree must be a **simple graph**.

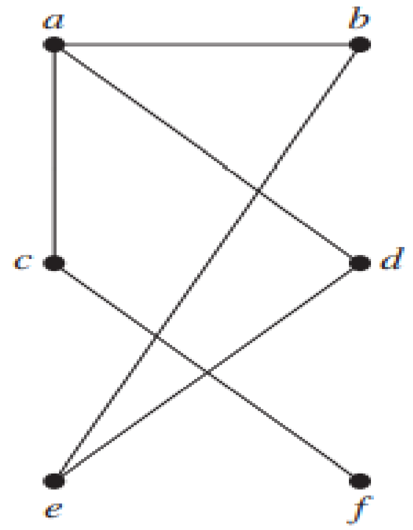
Check whether following graph are tree or not?



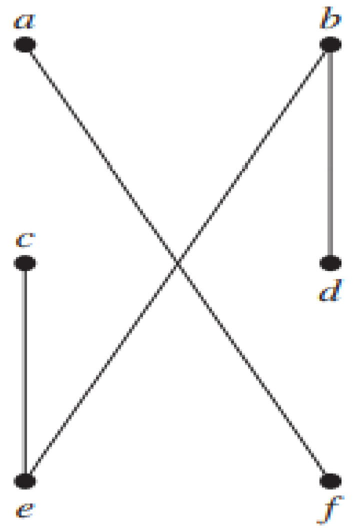
G_1



G_2



G_3



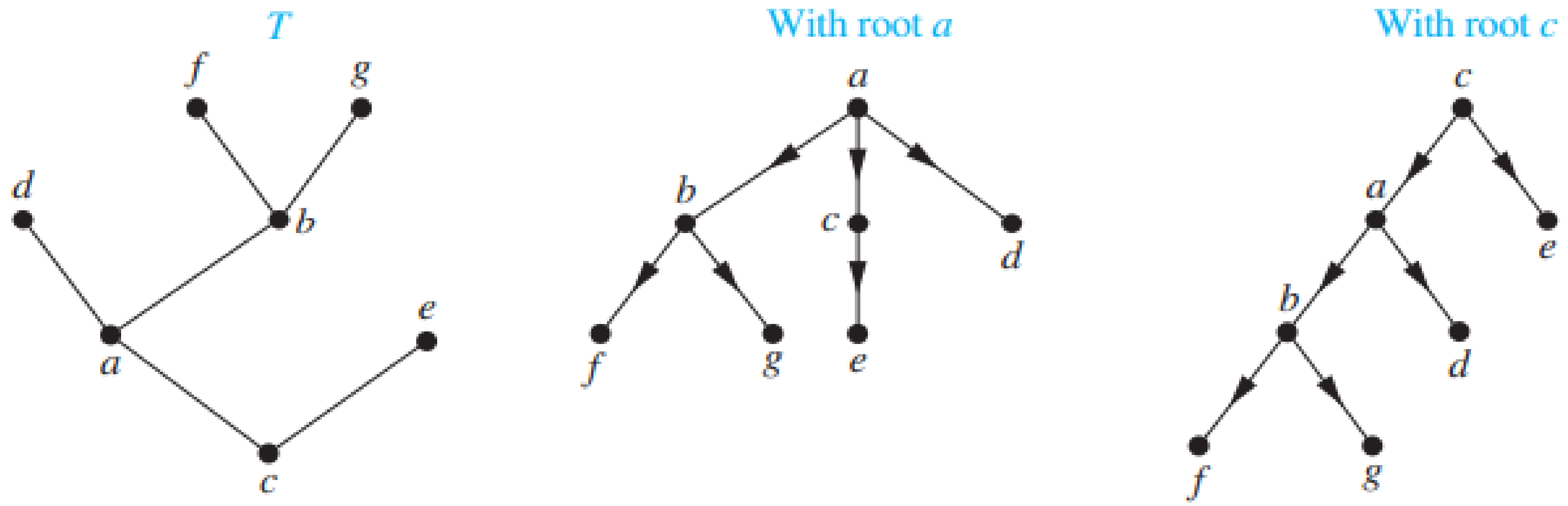
G_4

Solution…

- G1 and G2 are trees, because both are connected graphs with no simple circuits.
- G3 is not a tree because e, b, a, d, e is a simple circuit in this graph.
- Finally, G4 is not a tree because it is not connected.
- G4 is called forests and have the property that each of their connected components is a tree.

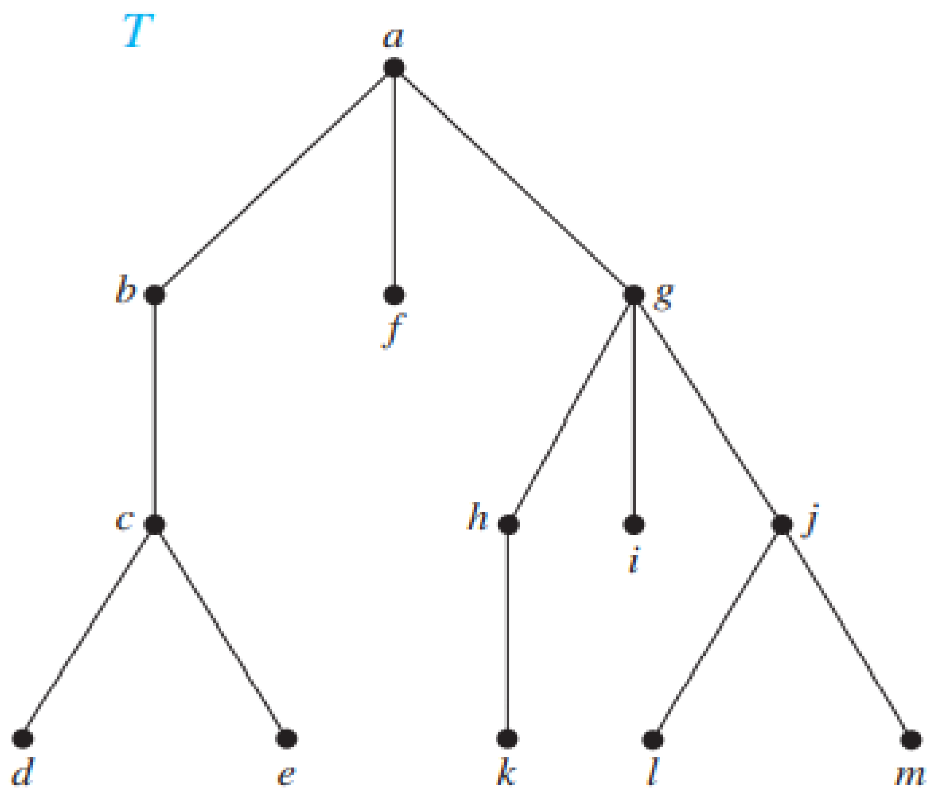
Contd…

- Tree can also be defined as: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.
- A **rooted tree** is a tree in which one vertex has been designated as the root and every edge is directed away from the root.



Rooted tree terminology

- Parent: the parent of v is the unique vertex u such that there is a directed edge from u to v. (b,c,g,h,j)
- Child: When u is the parent of v, v is called a child of u.
- Vertices with the same parent are called **siblings**. (d and e,l and m)

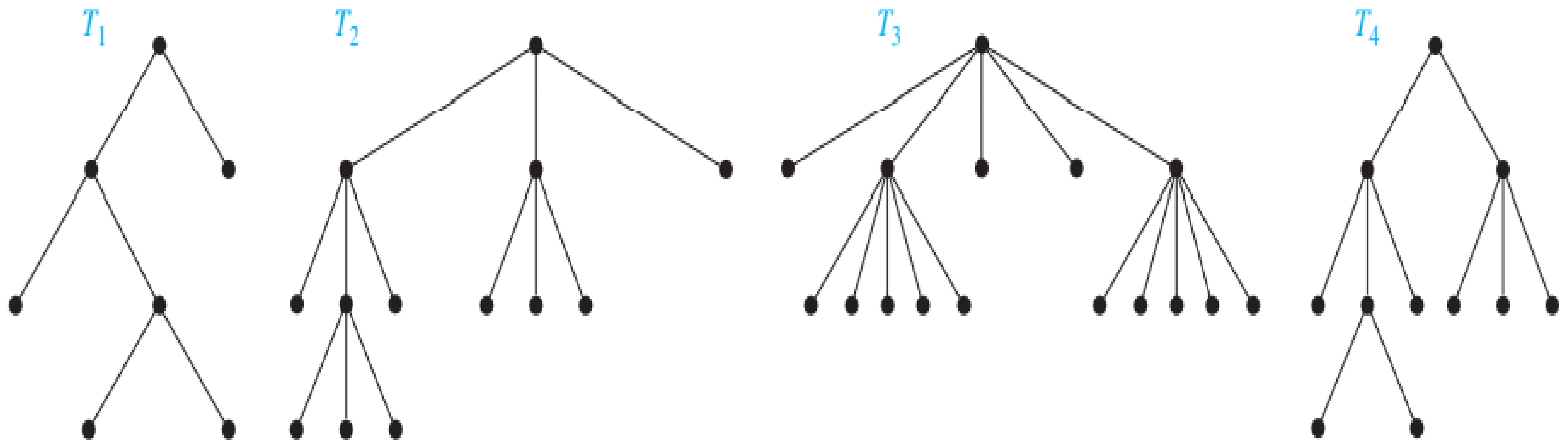


Contd…

- 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.
- The **descendants** of a vertex v are those vertices that have v as an ancestor.
- A vertex of a rooted tree is called a **leaf if it has no children**.
- Vertices that **have children are called internal vertices**.
- 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.

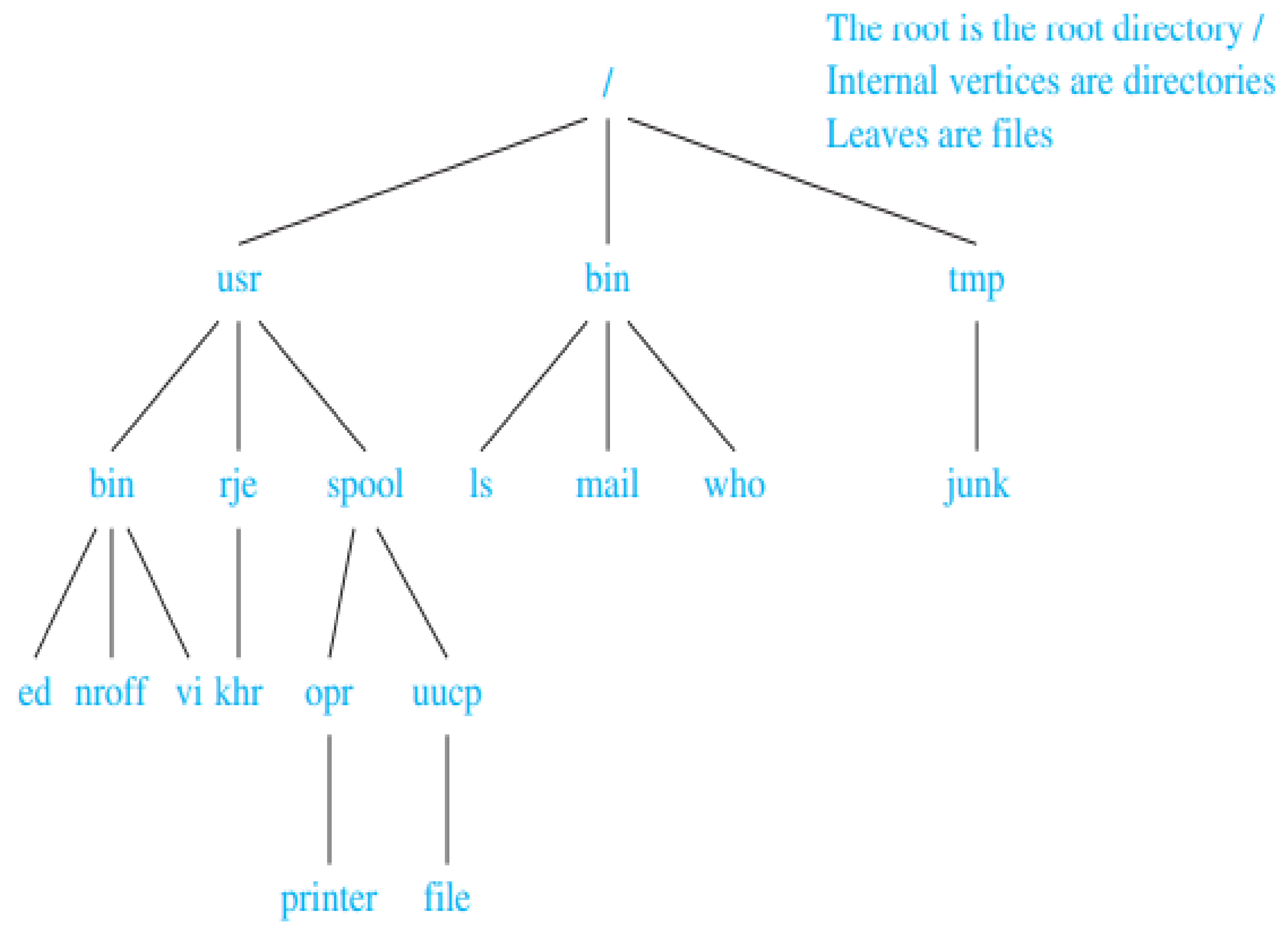
M-ary tree…

- A rooted tree is called an m-ary tree if every internal vertex has no more than m children.
- The tree is called a full m-ary tree if every internal vertex has exactly m children.
- An m-ary tree with m = 2 is called a binary tree.



T1: full 2-ary tree(binary)
T2: full 3-ary tree
T3: full 5ary tree
T4: not full 3ary-tree

Tree as a model for computer file system



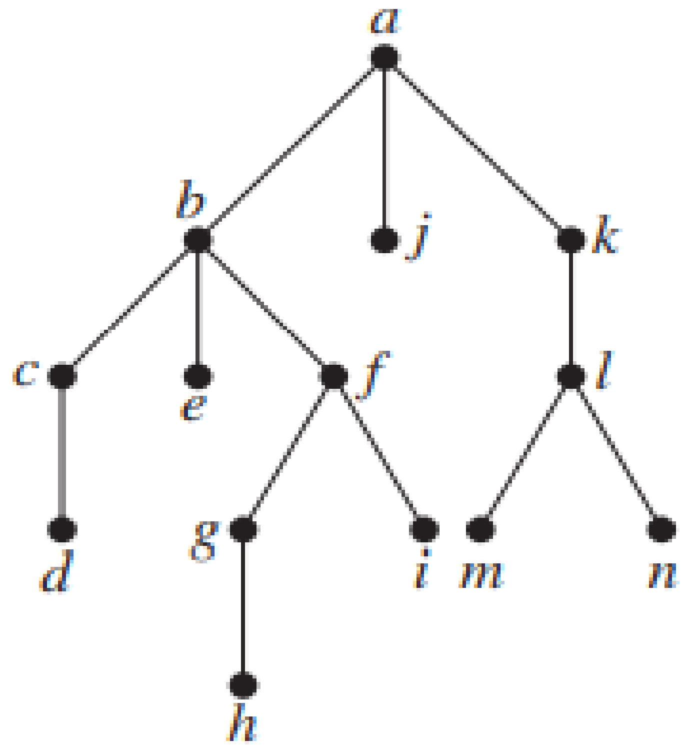
Contd…

- A tree with n vertices has n – 1 edges.
- Try for 5 vertices and try to connect with no cycle.
- A full m-ary tree with i internal vertices contains n = mi + 1 vertices.
- BALANCED m-ARY TREES: The level of a vertex v in a rooted tree is the length of the unique path from the root to this vertex. The level of the root is defined to be zero.
- The height of a rooted tree is the maximum of the levels of vertices. In other words, the height of a rooted tree is the length of the longest path from the root to any vertex

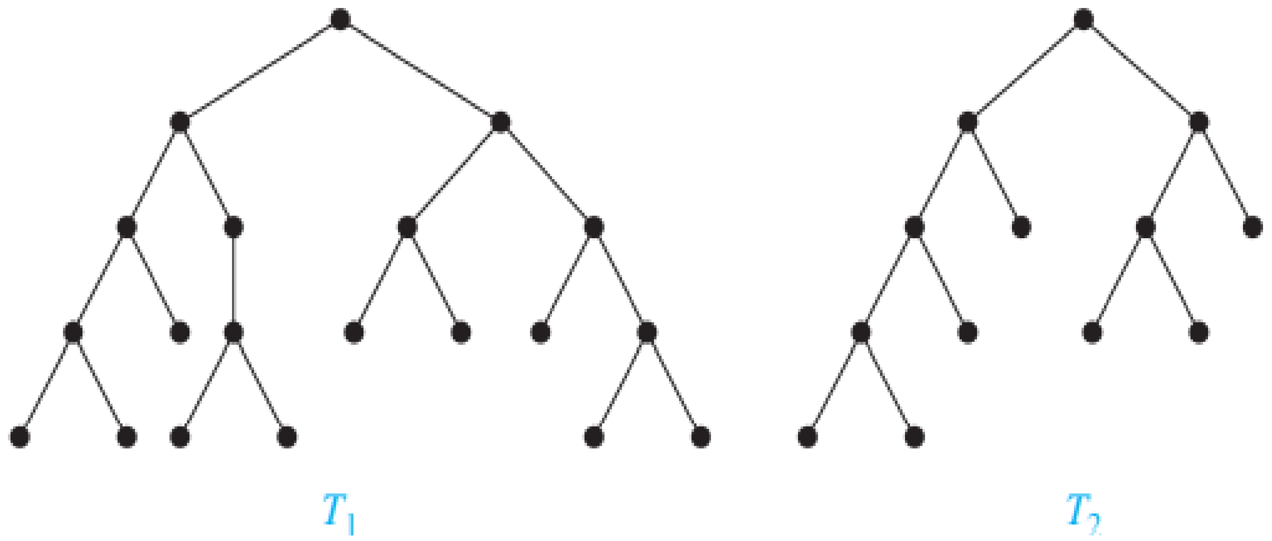
Balanced tree

- A rooted m-ary tree of height h is **balanced if all leaves are at levels h or h − 1.**
- The root a is at level 0.
- Vertices b, j , and k are at level 1.
- Vertices c, e, f , and l are at level 2.
- Vertices d, g, i, m, and n are at level 3.
- Finally, vertex h is at level 4.
- Because the largest level of any vertex is

4, this tree has height 4.



Which of the rooted trees shown in Figure are balanced?



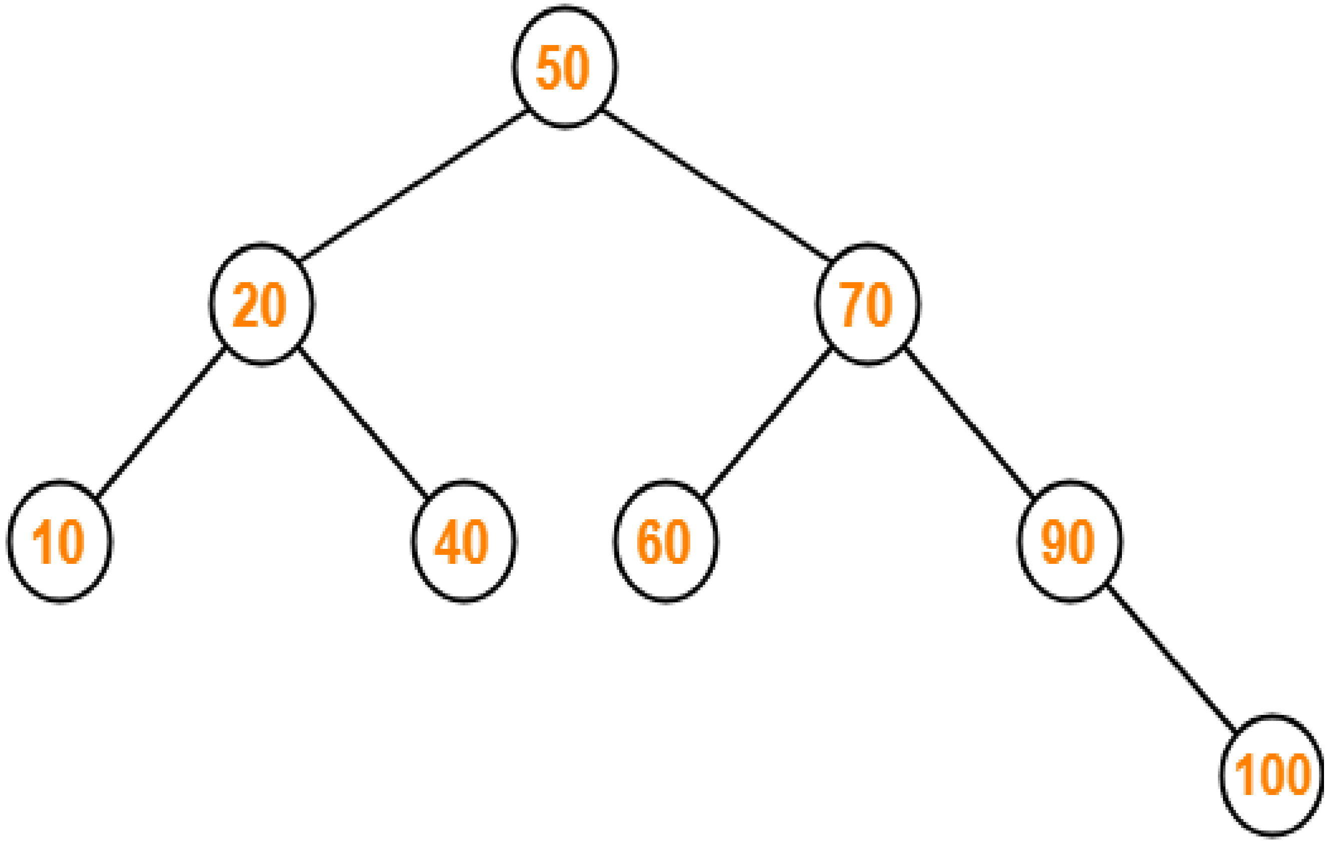
- T1 is balanced, because all its leaves are at levels 3 and 4.
- However, T2 is not balanced, because it has leaves at levels 2, 3, and 4.

Binary search tree

- How should items in a list be stored so that an item can be easily located?
- Searching for items in a list is one of the most important tasks that arises in computer science.
- Our primary goal is to implement a searching algorithm that finds items efficiently when the items are totally ordered.
- This can be implemented by binary search tree.
- In a binary tree, there can be at most 2 child.

Construct a binary search tree for 50, 70, 60, 20, 90, 10, 40, 100

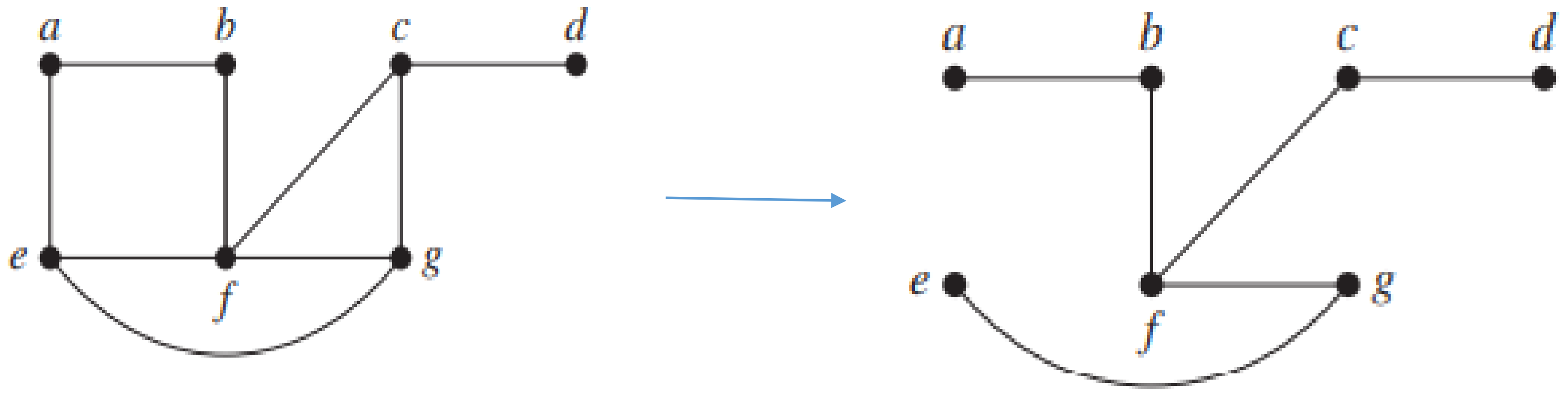
- In a binary search tree (BST), each node contains-
  - Only smaller values in its left sub tree
  - Only larger values in its right sub tree
- Node value are key.
- Always consider the first element as the root node.
- Consider the given elements and insert them in the BST one by one.



Binary search tree

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**.
- A simple graph with a spanning tree **must be connected**, because there is a path in the spanning tree between any two vertices.



Spanning tree

Assignment

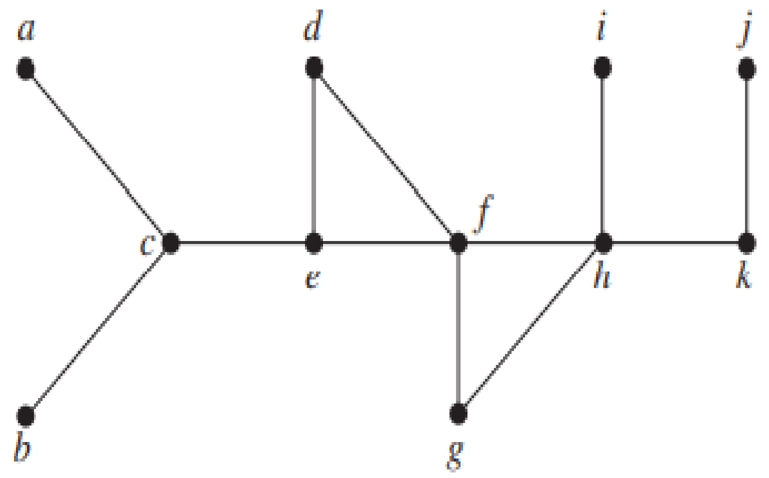
- Learn about IP multicasting as an example of spanning tree from page no. 786

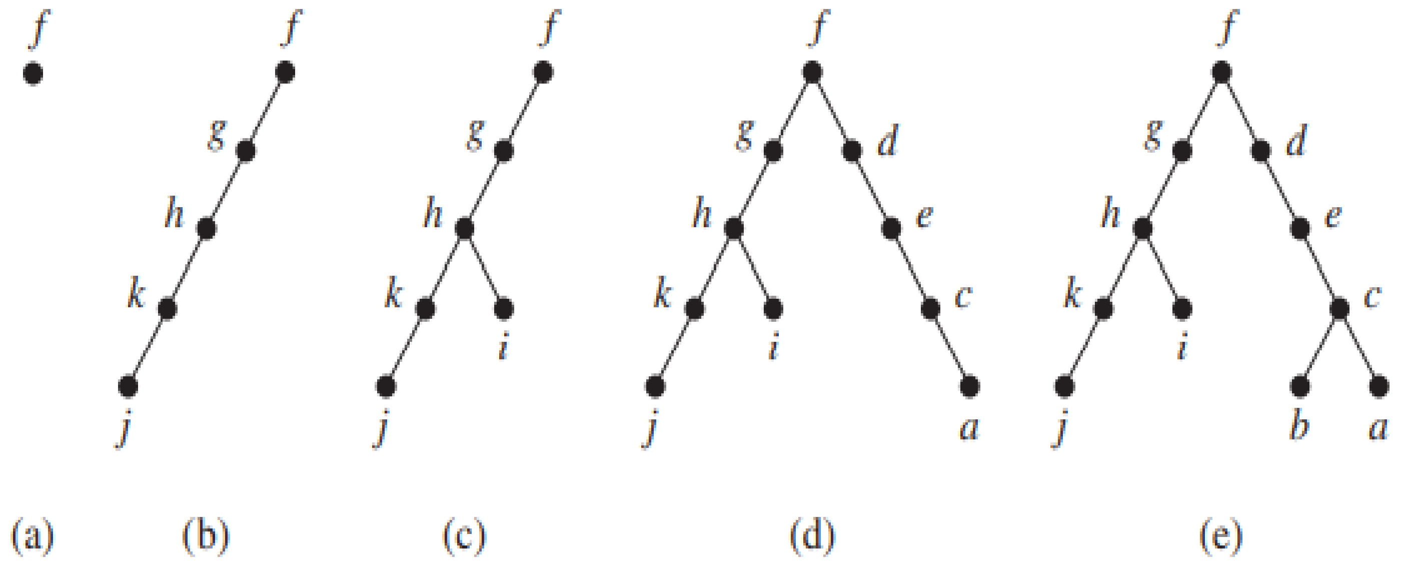
Depth first search/backtracking

- **Instead of constructing spanning trees by removing edges, spanning trees can be built up by successively adding edges.**
- Algorithm:
- Arbitrarily choose a vertex of the graph as the root.
- Form a path starting at this vertex by successively adding vertices and edges, where each new edge is incident with the last vertex in the path and a vertex not already in the path.
- Continue adding vertices and edges to this path as long as possible.
- If the path goes through all vertices of the graph, the tree consisting of this path is a spanning tree.

Algorithm contd…

- However, if the path does not go through all vertices, more vertices and edges must be added.
- Move back to the next to last vertex in the path, and, if possible, form a new path starting at this vertex passing through vertices that were not already visited.
- If this cannot be done, move back another vertex in the path, that is, two vertices back in the path, and try again.



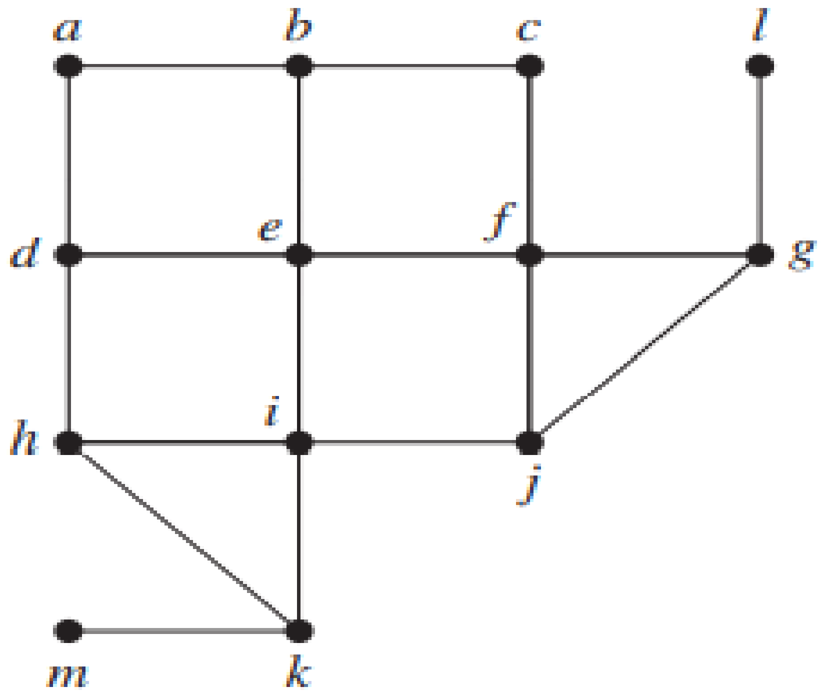


Spanning tree

Breadth first search

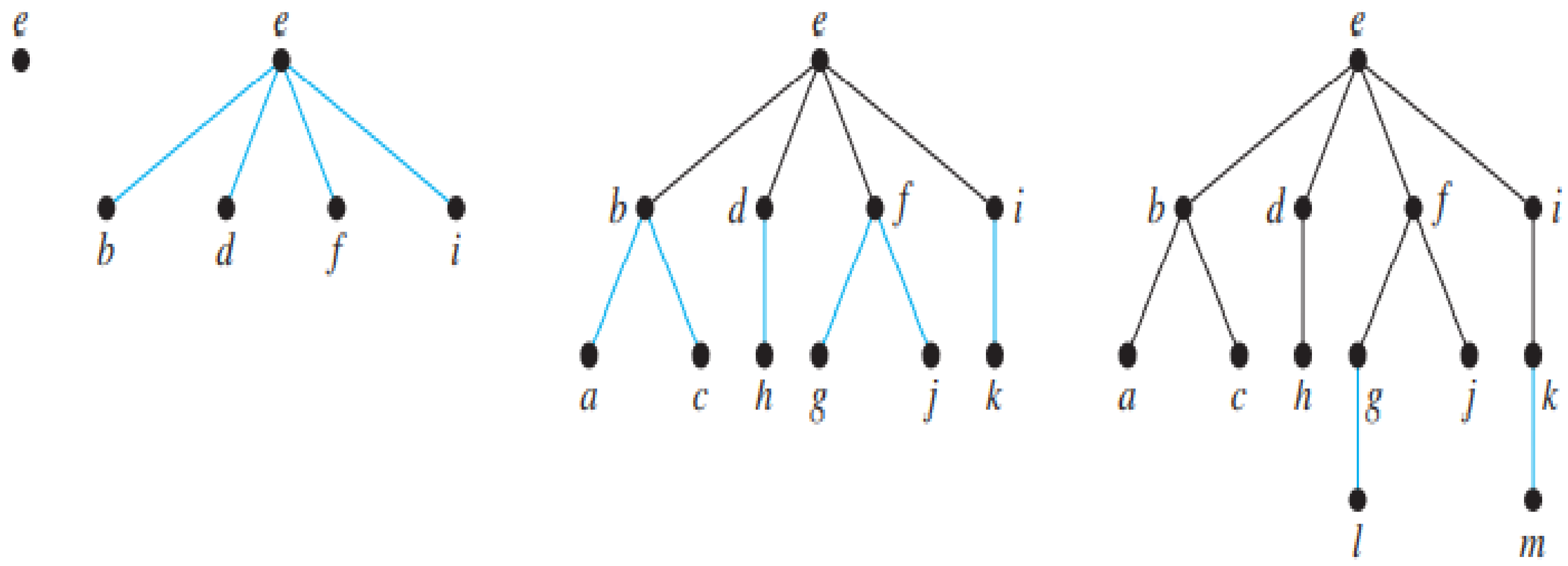
- Arbitrarily choose a root from the vertices of the graph. Then add all edges incident to this vertex.
- The new vertices added at this stage become the **vertices at level 1** in the spanning tree.
- Next, for each vertex at level 1, visited in order, add each edge incident to this vertex to the tree as long as it does not produce a simple circuit.
- Arbitrarily order the children of each vertex at level 1. This produces the **vertices at level 2 in the tree**.
- Follow the same procedure until all the vertices in the tree have been added. The procedure ends because there are only a finite number of edges in the graph.

Use breadth-first search to find a spanning tree for the graph shown below



Solution

- Solution: We choose the vertex e to be the root. Then we add edges incident with all vertices adjacent to e, so edges from e to b, d, f , and i are added.
- These four vertices are at level 1 in the tree. Next, add the edges from these vertices at level 1 to adjacent vertices not already in the tree.
- Hence, the edges from b to a and c are added, as are edges from d to h, from f to j and g, and from i to k.
- The new vertices a, c, h, j , g, and k are at level 2.
- Next, add edges from these vertices to adjacent vertices not already in the graph. This adds edges from g to l and from k to m.


e
e
b
d
f
i
e
b
d
f
i
a
c
h
g
j
k
e
b
d
f
i
a
c
h
g
j
k
l
m

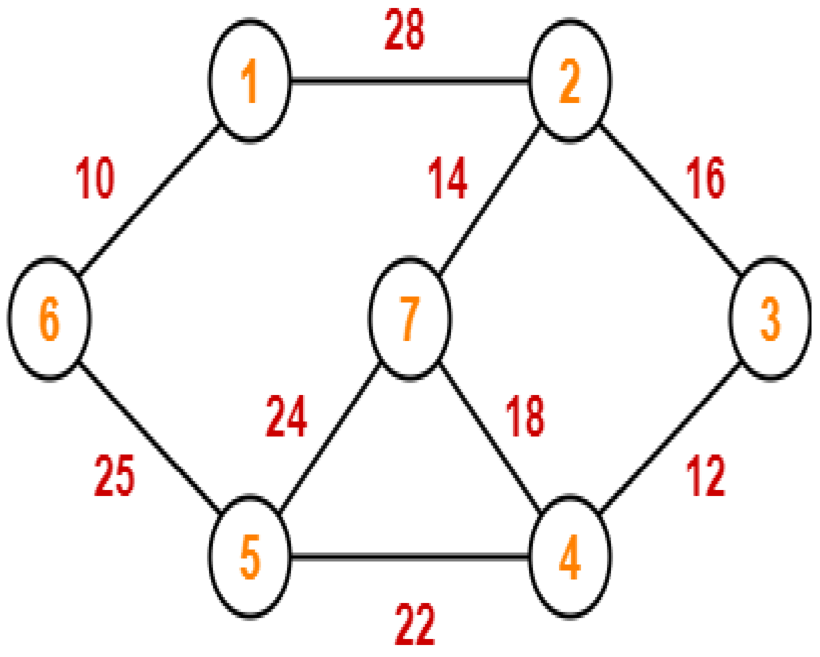
Minimum spanning tree

- A minimum spanning tree in a connected weighted graph is a spanning tree that has **the smallest possible sum of weights of its edges**.
- Two algorithms for constructing minimum spanning trees.
- Both are **greedy algorithms**. A greedy algorithm is a procedure that makes an optimal choice at each of its steps.
- Optimizing at each step does not guarantee that the optimal overall solution is produced.
- **However, the two algorithms presented in this section for constructing minimum spanning trees are greedy algorithms that do produce optimal solutions.**

Prims’s algorithm

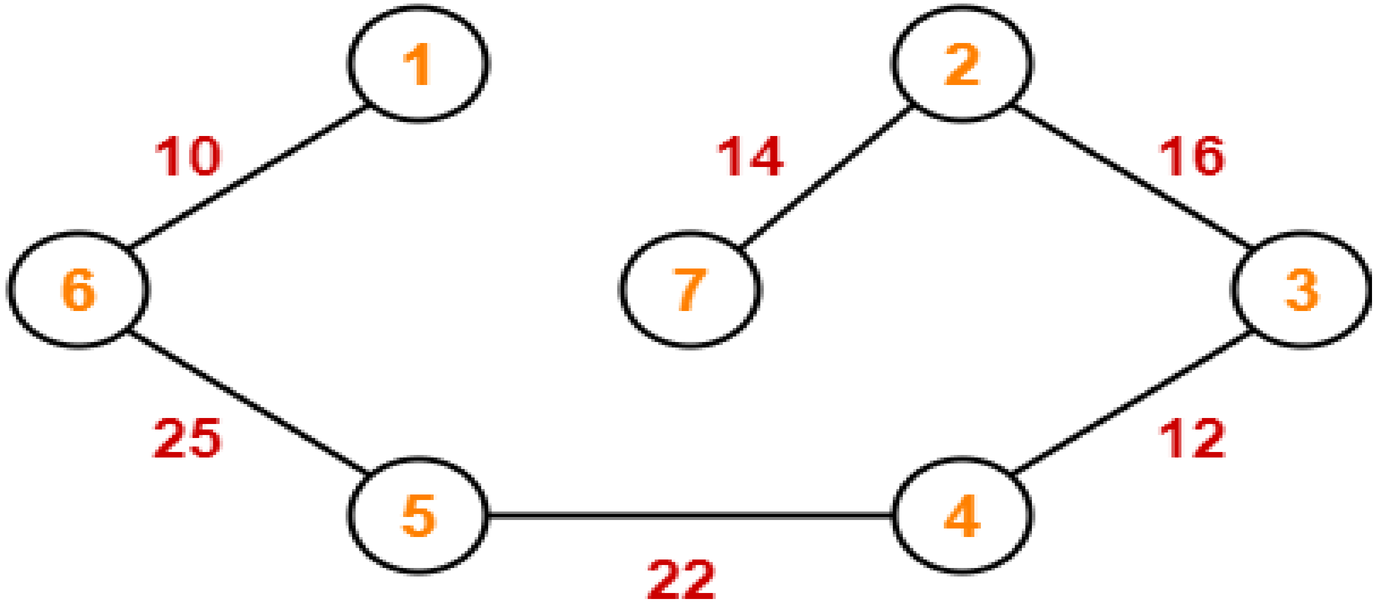
- Randomly choose any vertex.
- The vertex connecting to the edge having least weight is usually selected.
- Find all the edges that connect the tree to new vertices.(including old vertex)
- Find the least weight edge among those edges and include it in the existing tree.
- If including that edge creates a cycle, then reject that edge and look for the next least weight edge.
- Keep repeating until all the vertices are included and Minimum Spanning Tree (MST) is obtained.

Use prims algorithm to find MST for the given graph



Edge	Cost	To consider
(1,6)	10	(6,5),(1,2)
(6,5)	25	(1,2),(5,7),(5,4)
(5,4)	22	(1,2)(5,7)(4,7)(4,3)
(4,3)	12	(1,2)(5,7)(4,7)(3,2)
(3,2)	16	(5,7)(4,7)(2,7)
(2,7)	14	---------

Table for MST



Minimum spanning tree
10 + 25 + 22 + 12 + 16 + 14
= 99 units

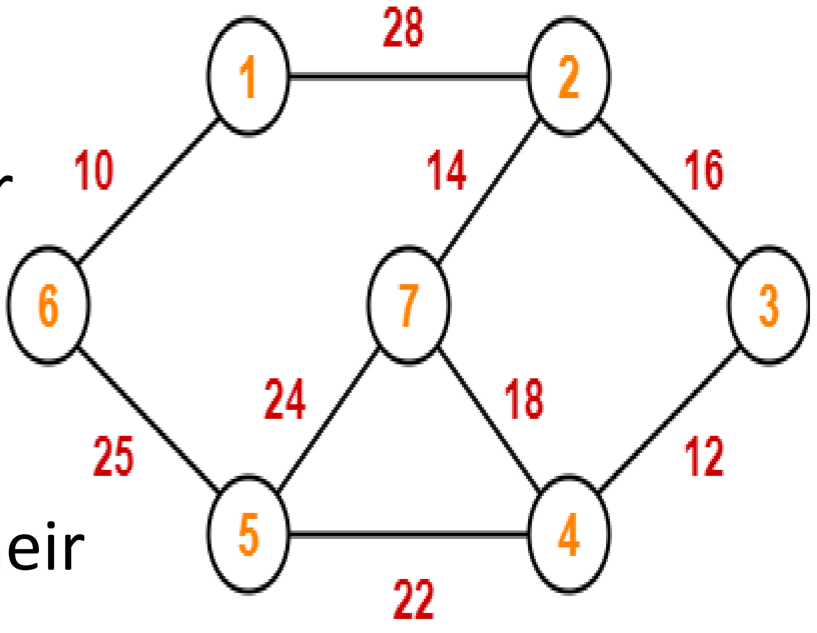
Kruskal algorithm

- Sort all the edges from low weight to high weight.
- Take the edge with the lowest weight and use it to connect the vertices of graph.
- If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.
- Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

Use kruskal algorithm to find MST for the given graph

Sorting the edges in ascending order
According to their weight
10,12,14,16,18,22,24,25,28

Connecting the edge according to their
Minimum weight
(6,1),(3,4),(7,2),(3,2),(4,5),(6,5)



trick

- To construct MST using Kruskal’s Algorithm,
- Simply draw all the vertices on the paper.
- Connect these vertices using edges with minimum weights such that no cycle gets formed.
- If all the edge weights are distinct, then both the algorithms are guaranteed to find the same MST.
- If all the edge weights are not distinct, then both the algorithms may not always produce the same MST.
- However, cost of both the MST_s would always be same in both the cases.