Graph Theory 9

Key Topics

Directed Graphs
Adirected Graphs
Incidence Matrix
Degree of Vertex
Walks and Paths
Hamiltonian Path
Graph Algorithms

9.1 Introduction

Graph theory is a practical branch of mathematics that deals with the arrangements of certain objects known as vertices (or nodes) and the relationships between them. It has been applied to practical problems such as the modelling of computer networks, determining the shortest driving route between two cities, the link structure of a website, the travelling salesman problem and the four-colour problem.[1]

Consider a map of the London underground, which is issued to users of the underground transport system in London. Then, this map does not represent every feature of the city of London, as it includes only material that is relevant to the users of the London underground transport system. In this map the exact geographical location of the stations is unimportant, and the essential information is how the stations are interconnected to one another, as this allows a passenger to plan a route

[1]The 4-colour theorem states that given any map it is possible to colour the regions of the map with no more than four colours such that no two adjacent regions have the same colour. This result was finally proved in the mid-1970s.

G. O'Regan, *Guide to Discrete Mathematics*, Texts in Computer Science,
DOI 10.1007/978-3-319-44561-8_9

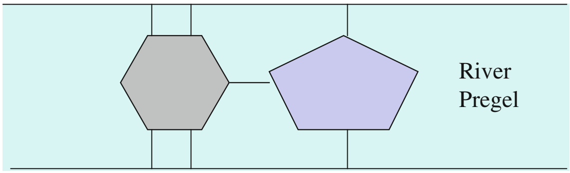


Fig. 9.1 Königsberg seven bridges problem

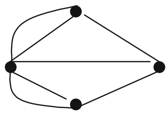


Fig. 9.2 Königsberg graph

from one station to another. That is, the map of the London underground is essentially a model of the transport system that shows how the stations are interconnected.

The seven bridges of Königsberg[2] (Fig. 9.1) is one of the earliest problems in graph theory. The city was set on both sides of the Pregel River in the early eighteenth century, and it consisted of two large islands that were connected to each other and the mainland by seven bridges. The problem was to find a walk through the city that would cross each bridge only once.

Euler showed that the problem had no solution, and his analysis helped to lay the foundations for graph theory as a discipline. This problem in graph theory is concerned with the question as to whether it is possible to travel along the edges of a graph starting from a vertex and returning to it and travelling along each edge exactly once. An Euler Path in a graph G is a simple path containing every edge of G.

Euler noted, in effect, that for a walk through a graph traversing each edge exactly once depends on the *degree* of the nodes (i.e. the number of edges touching it). He showed that a necessary and sufficient condition for the walk is that the graph is connected and has zero or two nodes of odd degree. For the Köningberg graph, the four nodes (i.e. the land masses) have odd degree (Fig. 9.2).

A *graph* is a collection of objects that are interconnected in some way. The objects are typically represented by vertices (or nodes), and the interconnections between them are represented by edges (or lines). We distinguish between directed

[2]Königsberg was founded in the thirteenth century by Teutonic knights and was one of the cities of the Hanseatic League. It was the historical capital of East Prussia (part of Germany), and it was annexed by Russia at the end of the Second World War. The German population either fled the advancing Red army or were expelled by the Russians in 1949. The city is now called Kaliningrad. The famous German philosopher, Immanuel Kant, spent all his life in the city, and is buried there.

and adirected graphs, where a *directed graph* is mathematically equivalent to a binary relation, and an *adirected (undirected) graph* is equivalent to a symmetric binary relation.

9.2 Undirected Graphs

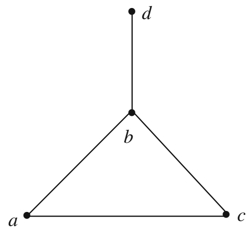
An *undirected graph* (*adirected graph*) (Fig. 9.3) G is a pair of finite sets (V, E) such that E is a binary symmetric relation on V. The set of vertices (or nodes) is denoted by $V(G)$ and the set of edges is denoted by $E(G)$.

A *directed graph* (Fig. 9.4) is a pair of finite sets (V, E) where E is a binary relation (that may not be symmetric) on V. A *directed acyclic graph* (*dag*) is a directed graph that has no cycles. The example below is of a directed graph with three edges and four vertices.

An edge $e \in E$ consists of a pair $\langle x, y \rangle$ where x, y are adjacent nodes in the graph. The *degree* of x is the number of nodes that are adjacent to x. The set of edges is denoted by $E(G)$, and the set of vertices is denoted by V(G).

A *weighted graph* is a graph $G = (V, E)$ together with a weighting function $w : E \rightarrow \mathbb{N}$, *which* associates a weight with every edge in the graph. A weighting function may be employed in modelling computer networks: for example, the weight of an edge may be applied to model the bandwidth of a telecommunications link between two nodes. Another application of the weighting function is in determining the distance (or shortest path) between two nodes in the graph (where such a path exists).

Fig. 9.3 Undirected graph



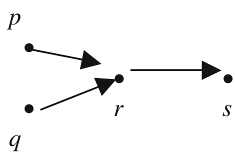
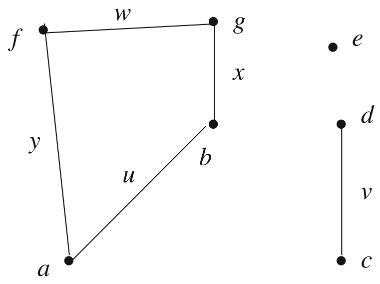


Fig. 9.4 Directed graph

For an adirected graph the weight of the edge is the same in both directions: i.e. $w(v_i, v_j) = w(v_j, v_i)$ for all edges $\langle v_i, v_j \rangle$ in the graph G, whereas the weights may be different for a directed graph.

Two vertices x, y are adjacent if $xy \in E$, and x and y are said to be incident to the edge xy. A matrix may be employed to represent the adjacency relationship.

Example 9.1



Consider the graph $G = (V, E)$ where $E = \{u = ab, v = cd, w = fg, x = bg, y = af\}$

An adjacency matrix (Fig. 9.5) may be employed to represent the relationship of adjacency in the graph. Its construction involves listing the vertices in the rows and columns, and an entry of 1 is made in the table if the two vertices are adjacent and 0 otherwise.

Similarly, we can construct a table describing the incidence of edges and vertices by constructing an incidence matrix (Fig. 9.6). This matrix lists the vertices and edges in the rows and columns, and an entry of 1 is made if the vertex is one of the nodes of the edge and 0 otherwise.

Fig. 9.5 Adjacency matrix

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	0	0	0	1	0
b	1	0	0	0	0	0	1
c	0	0	0	1	0	0	0
d	0	0	1	0	0	0	0
e	0	0	0	0	0	0	0
f	1	0	0	0	0	0	1
g	0	1	0	0	0	1	0

Fig. 9.6 Incidence matrix

	u	*v*	*w*	*x*	*y*
a	1	0	0	0	1
b	1	0	0	1	0
c	0	1	0	0	0
d	0	1	0	0	0
e	0	0	0	0	0
f	0	0	1	0	1
g	0	0	1	1	0

Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be isomorphic if there exists a bijection $f: V \rightarrow V'$ such that for any $u, v \in V$, $uv \in E$, $f(u)f(v) \in E'$. The mapping f is called an isomorphism. Two graphs that are isomorphic are essentially equivalent apart from a re-labelling of the nodes and edges.

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs then G' is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. Given $G = (V, E)$ and $V' \subseteq V$ then we can induce a subgraph $G' = (V', E')$ by restricting G to V' (denoted by $G\,|_{V'}|$). The set of edges in E′ is defined as:

$$E' = \{e \in E : e = uv \text{ and } u, v \in V'\}$$

The *degree* of a vertex v is the number of distinct edges incident to v. It is denoted by deg v where

$$\begin{aligned} \deg v &= |\{e \in E : e = vx \text{ for some } x \in V\}| \\ &= |\{x \in V : vx \in E\}| \end{aligned}$$

A vertex of degree 0 is called an isolated vertex.

Theorem 9.1 *Let $G = (V, E)$ be a graph then*

$$\sum\nolimits_{v \in V} \deg v = 2|E|$$

Proof This result is clear since each edge contributes one to each of the vertex degrees. The formal proof is by induction based on the number of edges in the graph, and the basis case is for a graph with no edges (i.e. where every vertex is isolated), and the result is immediate for this case.

The inductive step (strong induction) is to assume that the result is true for all graphs with k or fewer edges. We then consider a graph $G = (V, E)$ with $k + 1$ edges.

Choose an edge $e = xy \in E$ and consider the graph $G' = (V, E')$ where $E' = E \backslash \{e\}$. Then G' is a graph with k edges and therefore letting $\deg' v$ represents the degree of a vertex in G' we have:

$$\sum_{v \in V} \deg' v = 2|E'| = 2(|E| - 1) = 2|E| - 2$$

The degree of x and y are one less in G' than they are in G. That is,

$$\begin{aligned} &\sum_{v \in V} \deg v - 2 = \sum_{v \in V} \deg' v = 2|E| - 2 \\ &\Rightarrow \sum_{v \in V} \deg v = 2|E| \end{aligned}$$

A graph $G = (V, E)$ is said to be *complete* if all the vertices are adjacent: i.e. $E = V \times V$. A graph $G = (V, E)$ is said to be *simple graph* if each edge connects two different vertices, and no two edges connect the same pair of vertices. Similarly, a graph that may have multiple edges between two vertices is termed a *multigraph*.

A common problem encountered in graph theory is determining whether or not there is a route from one vertex to another. Often, once a route has been identified the problem then becomes that of finding the shortest or most efficient route to the destination vertex. A graph is said to be *connected* if for any two given vertices v_1, v_2 in V there is a path from v_1 to v_2.

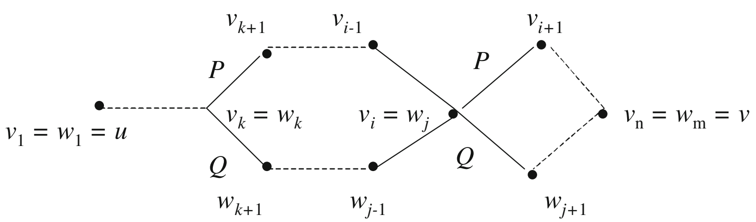
Consider a person walking in a forest from A to B where the person does not know the way to B. Often, the route taken will involve the person wandering around aimlessly, and often retracing parts of the route until eventually the destination B is reached. This is an example of a *walk* from v_1 to v_k where there may be repetition of edges.

If all of the edges of a walk are distinct then it is called a *trail*. A *path* $v_1, v_2, \ldots, v_k$ from vertex v_1 to v_k is of length $k - 1$ and consists of the sequence of edges $\langle v_1, v_2\rangle, \langle v_2, v_3\rangle, \ldots, \langle v_{k-1}, v_k\rangle$ where each $\langle v_i, v_{i+1}\rangle$ is an edge in E. The vertices in the path are all distinct apart from possibly v_1 and v_k. The path is said to be a cycle if $v_1 = v_k$. A graph is said to be *acyclic* if it contains no cycles.

Theorem 9.2 *Let $G = (V, E)$ be a graph and $W = v_1, v_2, \ldots, v_k$ be a walk from v_1 to, v_k. Then there is a path from v_1 to, v_k using only edges of W.*

Proof The walk W may be reduced to a path by successively replacing redundant parts in the walk of the form $v_i\ v_{i+1}\ \ldots,\ v_j$ where $v_i = v_j$ with v_i. That is, we successively remove cycles from the walk and this clearly leads to a path (not necessarily the shortest path) from v_1 to, v_k.

Theorem 9.3 *Let $G = (V, E)$ be a graph and let $u, v \in V$ with $u \neq v$. Suppose that there exists two different paths from u to v in G, then G contains a cycle.*



Suppose that $P = v_1, v_2, \ldots, v_n$ and $Q = w_1, w_2, \ldots, w_m$ are two distinct paths from u to v (where $u \neq v$), and $u = v_1 = w_1$ and $v = v_n = w_m$. Suppose P and Q are identical for the first k vertices (k could be 1), and then differ (i.e., $v_{k+1} \neq w_{k+1}$). Then Q crosses P again at $v_n = w_m$, and possibly several times before then. Suppose

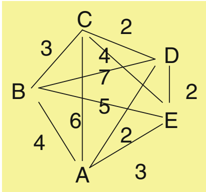


Fig. 9.7 Travelling salesman problem

the first occurrence is at $v_i = w_j$ with $k < i \leq n$. Then w_k, w_{k+1}, w_{k+2}, …, w_j v_{i-1}, v_{i-2}, …, v_k is a closed path (i.e. a cycle) since the vertices are all distinct.

If there is a path from v_1 to v_2 then it is possible to define the *distance* between v_1 and v_2. This is defined to be the total length (number of edges) of the shortest path between v_1 and v_2.

9.2.1 Hamiltonian Paths

A *Hamiltonian path*[3] in a graph $G = (V, E)$ is a path that visits every vertex once and once only. In other words, the length of a Hamiltonian path is $|V| - 1$. A graph is Hamiltonian-connected if for every pair of vertices there is a Hamiltonian path between the two vertices.

Hamiltonian paths are applicable to the travelling salesman problem, where a salesman[4] wishes to travel to k cities in the country without visiting any city more than once. In principle, this problem may be solved by looking at all of the possible routes between the various cities, and choosing the route with the minimal distance.

For example, Fig. 9.7 shows five cities and the connections (including distance) between them. Then, a travelling salesman starting at A would visit the cities in the order AEDCBA (or in reverse order ABCDEA) covering a total distance of 14.

However, the problem becomes much more difficult to solve as the number of cities increase, and there is no general algorithm for its solution. For example, for the case of ten cities, the total number of possible routes is given by 9! = 362,880, and an exhaustive search by a computer is feasible and the solution may be determined quite quickly. However, for 20 cities, the total number of routes is given by $19! = 1.2 \times 10^{17}$, and in this case it is no longer feasible to do an exhaustive search by a computer.

There are several sufficient conditions for the existence of a Hamiltonian path, and Theorem 9.4 describes a condition that is sufficient for the existence of a Hamiltonian path.

[3]These are named after Sir William Rowan Hamilton, a nineteenth century Irish mathematician and astronomer, who is famous for discovering quaternions [1].

[4]We use the term "salesman" to stand for "salesman" or "saleswoman".

Theorem 9.4 *Let $G = (V, E)$ be a graph with $|V| = n$ and such that deg v + deg $w \geq n - 1$ for all non-adjacent vertices v and w. Then G possesses a Hamiltonian path.*

Proof The first part of the proof involves showing that G is connected, and the second part involves considering the largest path in G of length $k - 1$ and assuming that $k < n$. A contradiction is then derived and it is deduced that $k = n$.

We assume that $G' = (V', E')$ and $G'' = (V'', E'')$ are two connected components of G, then $|V'| + |V''| \leq n$ and so if $v \in V'$ and $w \in V''$ then $n - 1 \leq$ deg v + deg $w \leq |V'| - 1 + |V''| - 1 = |V'| + |V''| - 2 \leq n - 2$ which is a contradiction, and so G must be connected.

Let $P = v_1, v_2, \ldots, v_k$ be the largest path in G and suppose $k < n$. From this a contradiction is derived, and the details for are in [2].

9.3 Trees

An acylic graph is termed a *forest* and a connected forest is termed a *tree*. A graph G is a tree if and only if for each pair of vertices in G there exists a unique path in G joining these vertices. This is since G is connected and acyclic, with the connected property giving the existence of at least one path and the acylic property giving uniqueness.

A *spanning tree* $T = (V, E')$ for the connected graph $G = (V, E)$ is a tree with the same vertex set V. It is formed from the graph by removing edges from it until it is acyclic (while ensuring that the graph remains connected).

Theorem 9.5 *Let $G = (V, E)$ be a tree and let $e \in E$ then $G' = (V, E\backslash\{e\})$ is disconnected and has two components.*

Proof Let $e = uv$ then since G is connected and acyclic uv is the unique path from u to v, and thus G' is disconnected since there is no path from u to v in G'.

It is thus clear that there are at least two components in G' with u and v in different components. We show that any other vertex w is connected to u or to v in G'.

Since G is connected there is a path from w to u in G, and if this path does not use e then it is in G' as well, and therefore u and w are in the same component of G'.

If it does use e then e is the last edge of the graph since u cannot appear twice in the path, and so the path is of the form $w, \ldots, v, u$ in G. Therefore, there is a path from w to v in G', and so w and v are in the same component in G'. Therefore, there are only two components in G'

Theorem 9.6 *Any connected graph $G = (V, E)$ possesses a spanning tree.*

Proof This result is proved by considering all connected subgraphs of ($G = V, E$) and choosing a subgraph T with $|E'|$ as small as possible. The final step is to show

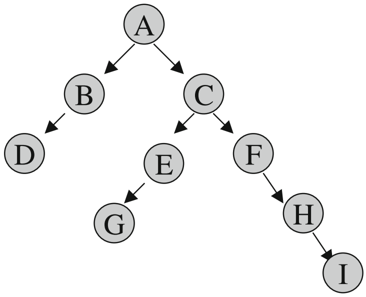


Fig. 9.8 Binary tree

that T is the desired spanning tree, and this involves showing that T is acyclic. The details of the proof are left to the reader.

Theorem 9.7 *Let* $G = (V, E)$ *be a connected graph, then* G *is a tree if and only if* $|E| = |V| - 1$.

Proof This result may be proved by induction on the number of vertices $|V|$ and the applications of Theorems 9.5 and 9.6.

9.3.1 Binary Trees

A *binary tree* (Fig. 9.8) is a tree in which each node has at most two child nodes (termed left and right child nodes). A node with children is termed a *parent node*, and the top node of the tree is termed the root node. Any node in the tree can be reached by starting from the root node, and by repeatedly taking either the left branch (left child) or right branch (right child) until the node is reached. Binary trees are used in computing to implement efficient searching algorithms. (We gave an alternative recursive definition of a binary tree in Chap. 4).

The *depth* of a node is the length of the path (i.e. the number of edges) from the root to the node. The depth of a tree is the length of the path from the root to the deepest node in the tree. A *balanced* binary tree is a binary tree in which the depth of the two subtrees of any node never differs by more than one. The root of the binary tree in Fig. 9.8 is A and its depth is 4. The tree is unbalanced and unsorted.

Tree traversal is a systematic way of visiting each node in the tree exactly once, and we distinguish between *breadth first search* in which every node on a particular level is visited before going to a lower level, and *depth first search* where one starts at the root and explores as far as possible along each branch before backtracking. The traversal in depth first search may be in preorder, inorder or postorder.

9.4 Graph Algorithms

Graph algorithms are employed to solve various problems in graph theory including network cost minimization problems; construction of spanning trees; shortest path algorithms; longest path algorithms; and timetable construction problems.

A length function $l : E \to \mathbb{R}$ may be defined on the edges of a connected graph $G = (V, E)$, and a shortest path from u to v in G is a path P with edge set E' such that $l(E')$ is minimal.

Due to space constraints it is not possible to describe graph algorithms in this section. The reader should consult the many texts on graph theory to explore many well-known graph algorithms such as Dijkstra's shortest path algorithm and longest path algorithm (e.g. as described in [2]). Kruskal's minimal spanning tree algorithm and Prim's minimal spanning tree algorithms are described in [2]. Next, we briefly discuss graph colouring in the next section.

9.5 Graph Colouring and Four-Colour Problem

It is very common for maps to be coloured in such a way that neighbouring states or countries are coloured differently. This allows different states or countries to be easily distinguished as well as the borders between them. The question naturally arises as to how many colours are needed (or determining the least number of colours needed) to colour the entire map, as it might be expected that a large number of colours would be needed to colour a large complicated map.

However, it may come as a surprise that in fact very few colours are required to colour any map. A former student of the British logician, Augustus De Morgan, had noticed this in the mid-1800s, and he proposed the conjecture of the four-colour theorem. There were various attempts to prove that four colours were sufficient from the mid-1800s onwards, and it remained a famous unsolved problem in mathematics until the late twentieth century.

Kempe gave an erroneous proof of the four-colour problem in 1879, but his attempt led to the proof that five colours are sufficient (which was proved by Heawod in the late 1800s). Appel and Haken of the University of Illinois finally provided the proof that 4 colours are sufficient in the mid-1970s (using over 1000 h of computer time in their proof).

Each map in the plane can be represented by a graph, with each region of the graph represented by a vertex. Edges connect two vertices if the regions have a common border. The colouring of a graph is the assignment of a colour to each vertex of the graph so that no two adjacent vertices in this graph have the same colour.

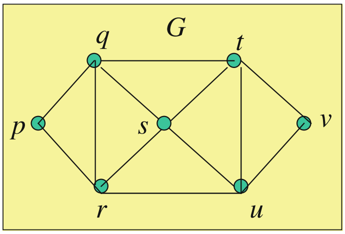


Fig. 9.9 Determining the chromatic colour of G

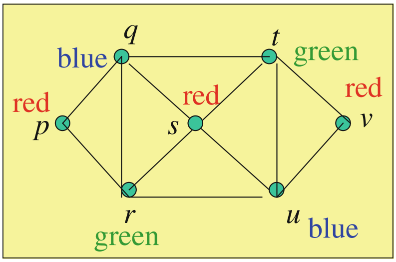


Fig. 9.10 Chromatic colouring of G

Definition
Let $G = (V, E)$ be a graph and let C be a finite set called the colours. Then, a colouring of G is a mapping $\kappa : V \rightarrow C$ such that if $uv \in E$ then $\kappa(u) \neq \kappa(v)$.

That is, the colouring of a simple graph is the assignment of a colour to each vertex of the graph such that if two vertices are adjacent then they are assigned a different colour. The chromatic number of a graph is the least number of colours needed for a colouring of the graph. It is denoted by χ(G).

Example 9.2 Show that the chromatic colour of the following graph G is 3 (this example is adapted from [3]) (Fig. 9.9).

Solution
The chromatic colour of G must be at least three since vertices p, q and r must have different colours, and so we need to show that three colours are in fact sufficient to colour G. We assign the colours red, blue and green to p, q and r, respectively. We immediately deduce that the colour of s must be red (as adjacent to q and r). From this, we deduce that t is coloured green (as adjacent to q and s) and u is coloured blue (as adjacent to s and t). Finally, v must be coloured red (as adjacent to u and t). This leads to the colouring of the graph G in Fig. 9.10.

Theorem 9.8 (Four-Colour Theorem) *The chromatic number of a planar graph G is less than or equal to* 4.

9.6 Review Questions

1. What is a graph and explain the difference between an adirected graph and a directed graph.
2. Determine the adjacency and incidence matrices of the following graph where $V = \{a, b, c, d, e\}$ and $E = \{ab, bc, ae, cd, bd\}$
3. Determine if the two graphs G and G' defined below are isomorphic.
4. $G = (V, E)$, $V = \{a, b, c, d, e, f, g\}$ and $E = \{ab, ad, ae, bd, ce, cf, dg, fg, bf\}$
5. $G' = (V', E')$, $V' = \{a, b, c, d, e, f, g\}$ and $\mathrm{E}' = \{ab, bc, cd, de, ef, fg, ga, ac, be\}$
6. What is a binary tree? Describe applications of binary trees.
7. Describe the travelling salesman problem and its applications.
8. Explain the difference between a walk, trail and path.
9. What is a connected graph?
10. Explain the difference between an incidence matrix and an adjacency matrix.
11. Complete the details of Theorems 9.6 and 9.7.
12. Describe the four-colour problem and its applications.

9.7 Summary

This chapter provided a brief introduction to graph theory, which is a practical branch of mathematics that deals with the arrangements of vertices and the edges between them. It has been applied to practical problems such as the modelling of computer networks, determining the shortest driving route between two cities, and the travelling salesman problem.

The seven bridges of Königsberg is one of the earliest problems in graph theory, and it was concerned with the problem was of finding a walk through the city that would cross each bridge once and once only. Euler showed that the problem had no solution, and his analysis helped to lay the foundations for graph theory.

An undirected graph G is a pair of finite sets (V, E) such that E is a binary symmetric relation on V, whereas a directed graph is a binary relation that is not symmetric. An adjacency matrix is used to represent whether two vertices are adjacent to each other, whereas an incidence matrix indicates whether a vertex is part of a particular edge.

A Hamiltonian path in a graph is a path that visits every vertex once and once only. Hamiltonian paths are applicable to the travelling salesman problem, where a salesman wishes to travel to k cities in the country without visiting any city more than once.

Graph colouring arose to answer the question as to how many colours are needed to colour an entire map. It may be expected that many colours would be required, but the four-colour theorem demonstrates that in fact four colours are sufficient to colour a planar graph.

A tree is a connected and acylic graph, and a binary tree is a tree in which each node has at most two child nodes.

References

1. Mathematics in Computing. Second Edition, Gerard O' Regan. Springer. 2012.
2. Discrete Mathematics. An Introduction for Software Engineers. Mike Piff. Cambridge University Press. 1991.
3. Discrete Mathematics and its Applications. 7th Edition. Kenneth H. Rosen Mc Graw Hill. 2012.