Graphs, trees and binary trees

1. Graphs
2. Trees
3. Binary trees

1. Graphs

A directed graph or digraph G is usually presented as a pair (V, E):
V is the vertex set and E is the edge set.

The elements of V are called vertices.

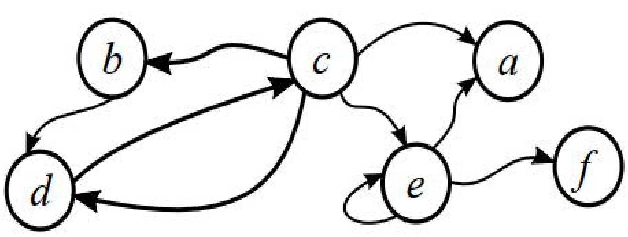
An element of E is called an edge and it is an ordered pair (a, b) of vertices.

For an ordered pair $(a, b) \neq (b, a)$.

Synonyms:

- vertex = node
- edge = arc = link

Example



$V = \{a, b, c, d, e, f\}$

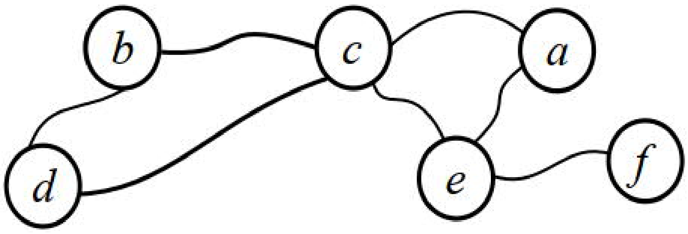
$E = \{(b, d), (d, c), (c, b), (c, d), (c, a), (c, e), (e, a), (e, e), (e, f)\}$

$\square$

If the graph edges have no direction, they we have an undirected graph.
For an undirected graph $(a, b) = (b, a)$.

NOTE: multiple edges not allowed

Example



$V = \{a, b, c, d, e, f\}$

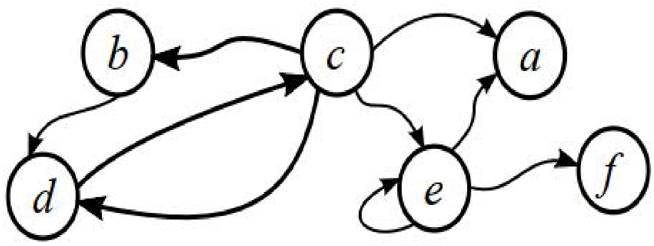
$E = \{(b, d), (d, c), (c, b), (c, a), (c, e), (e, a), (e, F)\}$

□

Terminology

- If edge (a, b) exists in a digraph, then b is adjacent to a.
- If edge (a, b) exists in an undirected graph, then b is adjacent to a and a is adjacent to b.
- If edge (a, b) exists in a digraph, then a is starting vertex and b is the final vertex.
- In an undirected graph the degree $a \in V$ is the number of vertices that are adjacent to a.
- An isolated vertex in an undirected graph is one whose degree is 0.
- In a digraph, the out-degree of $a \in V$ is the number of edges leaving from a and the in-degree of $a \in V$ is the number of edges entering a.

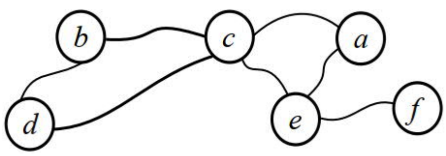
Example



vertex x	vertices adjacent to x	vertices to which x is adjacent	in-degree	out-degree
a	–	c, e	2	0
b	d	c	1	1
c	b, d, e, a	d	1	4
d	c	b, c	2	1
e	e, a, f	c, e	2	3
f	–	e	1	0

□

Example



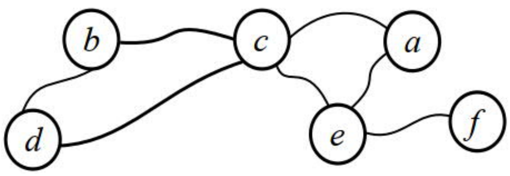
vertex x	a	b	c	d	e	f
vertices adjacent to x	c, e	d c	a, e, b, d	b, c	c, a, f	e
degree	2	2	4	2	3	1

The graph has no isolated vertices. ☐

More terminology

- A path of length k from vertex a_0 to vertex a_k is an ordered sequence of vertices $\langle a_0, a_1, a_2, \ldots a_k \rangle$ such that each edge (a_i, a_{i+1}), $i = 0, 1, \ldots k-1$ exists in the graph.
- In a simple path no vertex is repeated.
- If there is a path from a to b, then b is reachable from a.
- An undirected graph is connected if every vertex is reachable from every other vertex.
- A digraph is strongly connected if every vertex is reachable from every other vertex.
- The path $\langle a_0, a_1, a_2, \ldots a_k \rangle$ is a cycle when $a_0 = a_k$.
- In a simple cycle $\langle a_0, a_1, a_2, \ldots a_k \rangle$ the only repeated vertex is a_0.
- An acyclic graph has no cycles.

Example



(i) Paths from b to a

$\langle b, c, a \rangle$ length = 2

$\langle b, d, c, e, a \rangle$ length = 4

$\langle b, d, c, b, d, c, a \rangle$ length = 6

(ii) Simple paths from b to a

⟨b, c, a⟩ and ⟨b, d, c, e, a⟩

(iii) Every vertex is reachable from every other vertex. Hence the graph is connected.

(iv) Cycles in the graph

⟨b, d, c, b⟩ simple

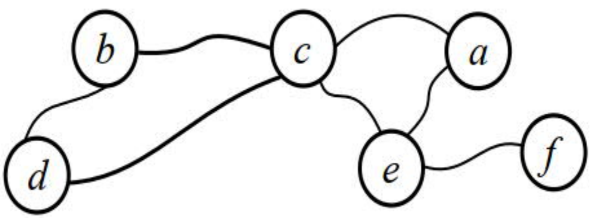
⟨a, c, e, a⟩ simple

⟨a, c, b, d, c, e, a⟩ not simple

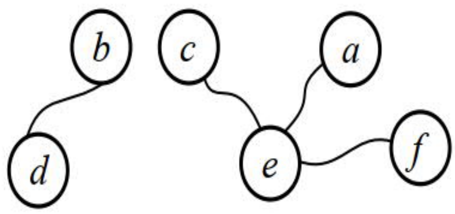
□

Example

G_1



G_2



G_1 is not acyclic.

G_2 is acyclic.

□

2. Trees

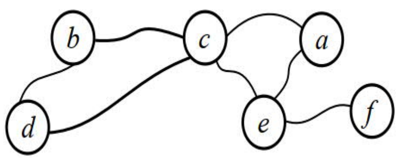
Undirected trees

Each of the following defines an undirected tree $G = (V, E)$:

- G is connected and acylic.
- G is connected and the number of edges is one less than the number of vertices.
- There is a unique simple path connecting every two vertices in G.
- G is acyclic, but adding any edge to E results in a graph with one cycle.

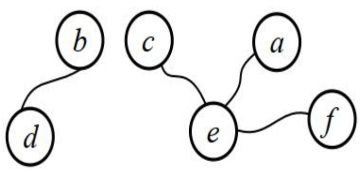
Example

G_1



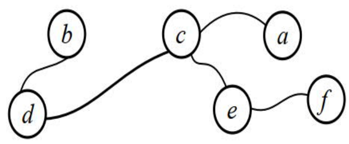
not acyclic, so not a tree

G_2



no path from *d* to *e*, so not a tree

G_3



is a tree

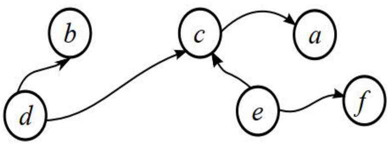
□

Rooted trees

A rooted tree is a digraph (V, E) where there is a unique simple path from one particular vertex, the root or r, to any other vertex, but there is no path from any vertex to r.

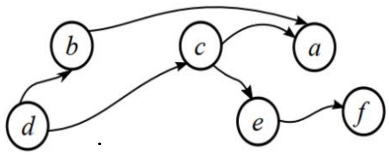
Example

D_1



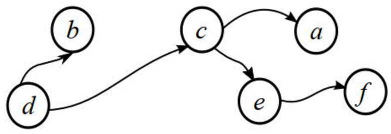
no vertex has a path
to all other vertices, so
not a rooted tree

D_2



two paths from *d* to *a*,
so not a rooted tree

D_3



is a rooted tree; root is *d*

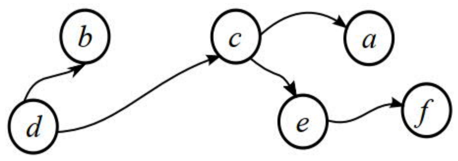
□

NOTE: usually omit arrows when drawing rooted tree and root is at the top

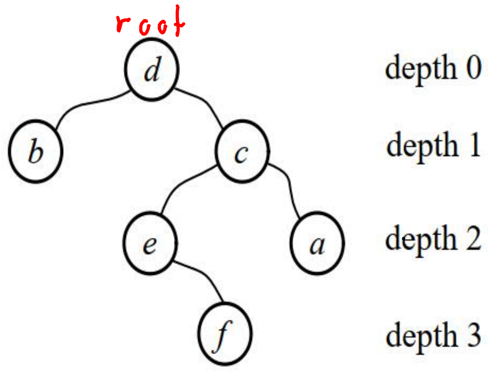
Terminology

- If a is a vertex on the unique path from r to b, then a is an ancestor of b.
- If a is an ancestor of b, then b is a descendant of a.
- The subtree rooted at a is a rooted tree having a as it's root and includes all descendants of a.
- If (a, b) is an edge, then a is b's parent and b is a' child.
- Two vertices having the same parent are siblings.
- A vertex with no children is a leaf or an external vertex.
- A non-leaf vertex is an internal vertex.
- The depth of vertex a is the length of the simple path from r to a.
- The tree's height is largest depth of any vertex.

Example



OR



(i) path from d to f : $\langle d, c, e, f \rangle$
From this path we observe
- c is an ancestor of f
- f is a descendant of c

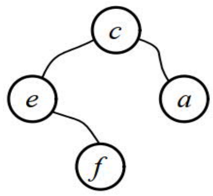
(ii) From edge (c, e)
c is e's parent and e is c s' child

(iii) Vertices e and a are siblings.
Vertices b and a are not siblings.

(iv) Vertices b, f and a are leaves.
Vertices d, e and c are internal vertices.

(v) The tree's height is 3.

(vi) The subtree rooted at c is



□

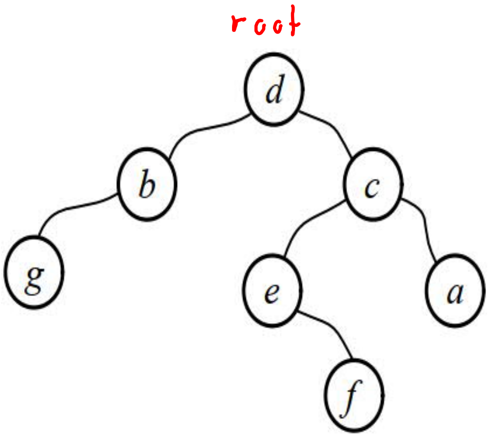
3. Binary trees

A binary tree is a rooted tree where every vertex has at most two children.

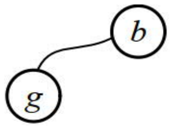
Terminology

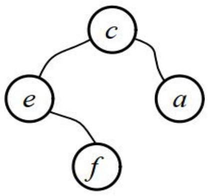
- The binary tree with no vertices is the empty tree or the null tree.
- Each child of a vertex is either the left child or the right child.
- When a vertex has no left (right) child, then the left (right) child is missing.
- If b is the left (right) child of a, then the left subtree (right subtree) of a is the subtree rooted at b.
  NOTE: both left and right subtrees are also binary trees
- In a full binary tree each vertex has either two children or no children.
- A complete binary tree is a full binary tree where all leaves are at the same depth.

Example



vertex	a	b	c	d	e	f	g
left child	–	g	e	b	–	–	–
right child	–	–	a	c	f	–	–
parent	c	d	d	–	c	e	b

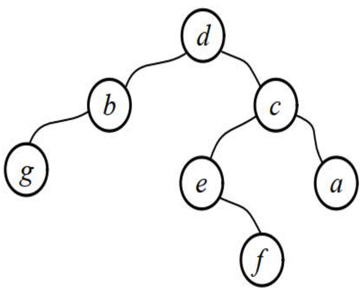
Left subtree of d: 

Right subtree of d: 

□

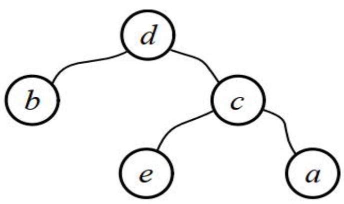
Example

T_1



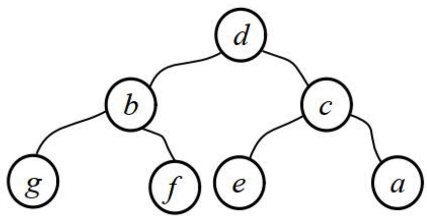
b has only one child,
so not a full binary tree

T_2



is a full binary tree,
but not complete since
leaves b and e are not at
same depth

T_3



is a complete binary tree

□