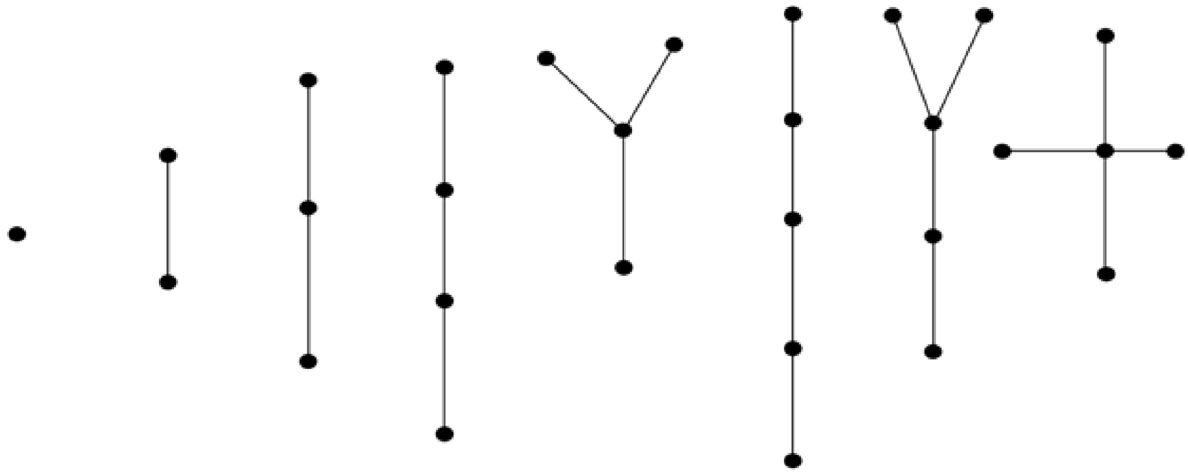
What is Tree and Forest?

Tree

- In graph theory, a **tree** is an **undirected, connected and acyclic graph**. In other words, a connected graph that does not contain even a single cycle is called a tree.
- A tree represents hierarchical structure in a graphical form.
- The elements of trees are called their nodes and the edges of the tree are called branches.
- A tree with n vertices has (n-1) edges.
- Trees provide many useful applications as simple as a family tree to as complex as trees in data structures of computer science.
- A **leaf** in a tree is a vertex of degree 1 or any vertex having no children is called a leaf.

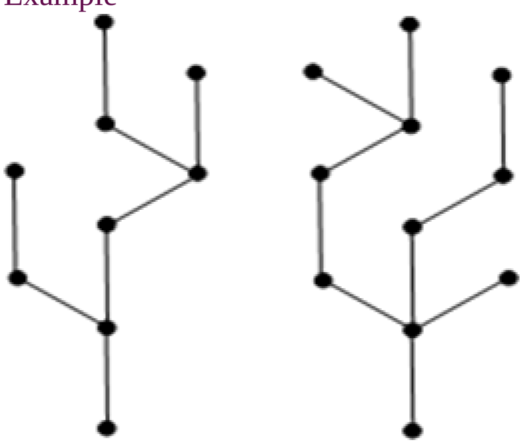


In the above example, all are trees with fewer than 6 vertices.

Forest

In graph theory, a **forest** is **an undirected, disconnected, acyclic graph**. In other words, a disjoint collection of trees is known as forest. Each component of a forest is tree.

Example



The above graph looks like a two sub-graphs but it is a single disconnected graph. There are no cycles in the above graph. Therefore it is a forest.

Properties of Trees

1. Every tree which has at least two vertices should have at least two leaves.
2. Trees have many characterizations:
   Let T be a graph with n vertices, then the following statements are equivalent:

- T is a tree.
- T contains no cycles and has n-1 edges.
- T is connected and has (n -1) edge.
- T is connected graph, and every edge is a cut-edge.
- Any two vertices of graph T are connected by exactly one path.
- T contains no cycles, and for any new edge e, the graph T+ e has exactly one cycle.

3. Every edge of a tree is cut -edge.
4. Adding one edge to a tree defines exactly one cycle.
5. Every connected graph contains a spanning tree.
6. Every tree has at least two vertices of degree two.

Which of the following graphs are trees?
G=(V,E) with V={a,b,c,d,e} and E={{a,b},{a,e},{b,c},{c,d},{d,e}}
G=(V,E) with V={a,b,c,d,e} and E={{a,b},{b,c},{c,d},{d,e}}
G=(V,E) with V={a,b,c,d,e} and E={{a,b},{a,c},{a,d},{a,e}}
G=(V,E) with V={a,b,c,d,e} and E={{a,b},{a,c},{d,e}}
Solution:
This is not a tree since it contains a cycle. Note also that there are too many edges to be a tree, since we know that all trees with v vertices have v−1 edges.

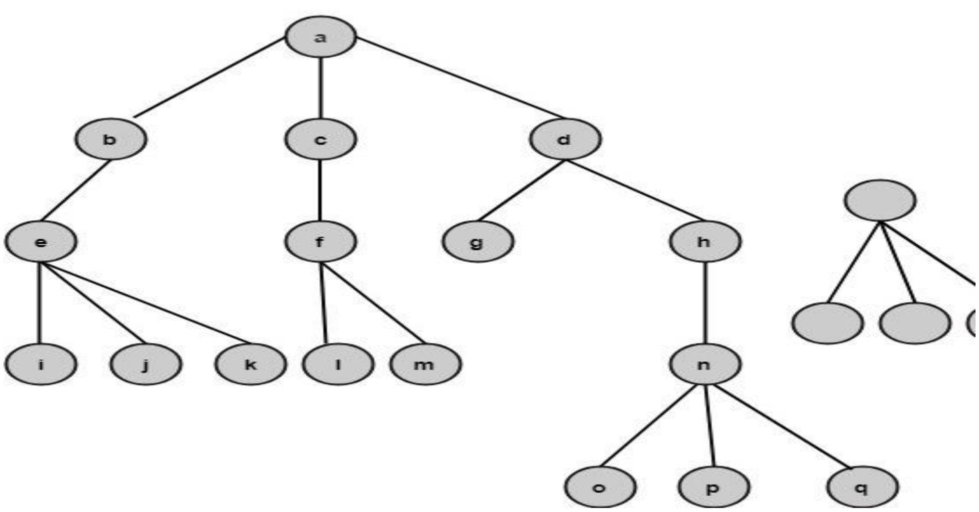
This is a tree since it is connected and contains no cycles (which you can see by drawing the graph). All paths are trees.

This is a tree since it is connected and contains no cycles (draw the graph). All stars are trees.

This is a not a tree since it is not connected. Note that there are not enough edges to be a tree.

Path length of a Vertex:

The path length of a vertex in a rooted tree is defined to be the number of edges in the path from the root to the vertex.
Example: Find the path lengths of the nodes b, f, l, q as shown in fig:

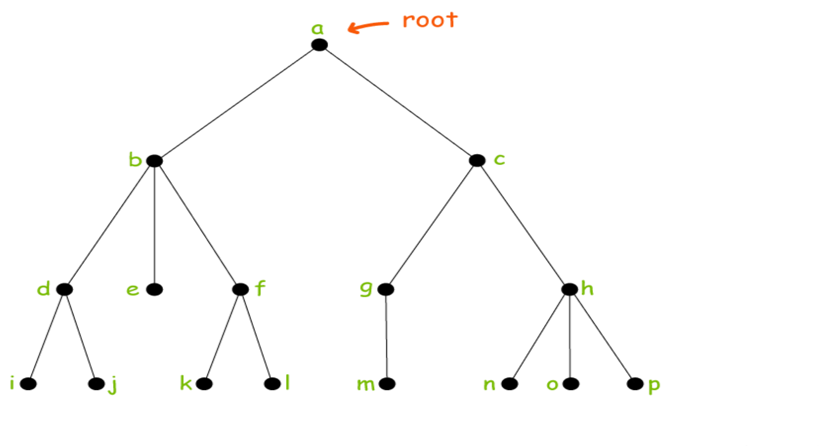


Solution: The path length of node b is one.
The path length of node f is two.
The path length of node l is three
The path length of the node q is four.

Rooted Tree

Now a **rooted tree** is a tree in which one vertex has been designated as the root, and every edge is directly away from the root. Think of a family tree where the oldest known relative is at the top of the tree, and all descendants are branched down from this one ancestor. This is the very idea of a rooted tree in graph theory.

Below is an example of a rooted tree and will help to highlight some of the critical vocabularies such as ancestors, descendants, parents, children, siblings, internal vertices, and leaves.

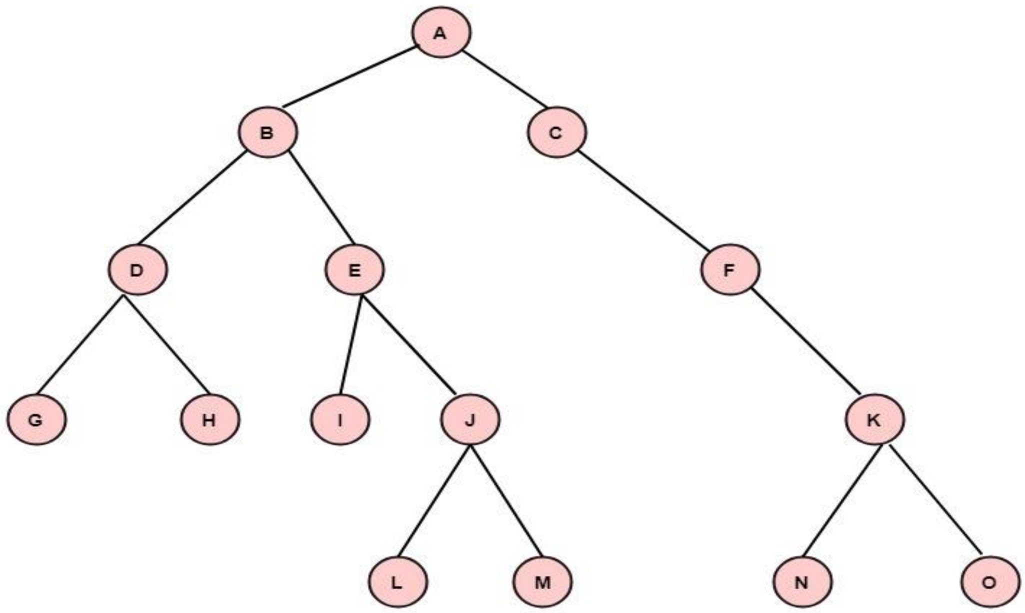


Rooted Tree Graph

- Children of a: b and c
- Parent of b and c: a
- Children of b: d, e and f (d, e and f are siblings)
- Children of c: g and h (g and h are siblings)
- Parent of d, e, and f: b
- Parent of g and h: c
- Leaves: e, i, j, k, l, m, n, o, p (vertices that do not have children)
- Ancestors of m: g, c, and a (all vertices in the path from the root to the desired vertex)
- Descendants of c: g, h, m, n, o, p (all vertices originating from the indicated vertex)
- Internal vertices: b, c, d, f, g, h (vertices that have children)

Example: For the tree as shown in fig:

- Which node is the root?
- Which nodes are leaves?
- Name the parent node of each node

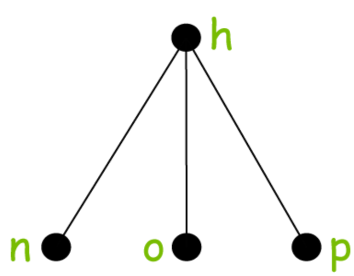


Solution: (i) The node A is the root node.
(ii) The nodes G, H, I, L, M, N, O are leaves.
(iii)

Nodes	**Parent**
B, C	A
D, E	B
F	C
G, H	D
I, J	E
K	F
L, M	J
N, O	K

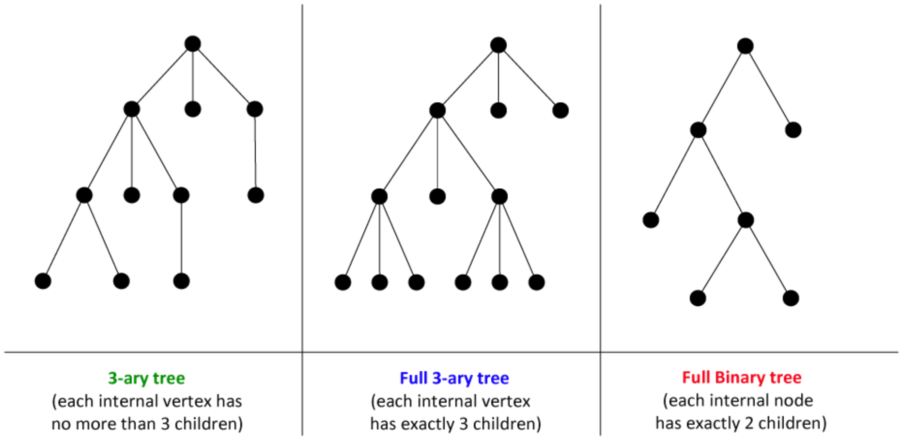
Subtree

And a **subtree** is a subgraph of a tree consisting of the indicated vertex and all of its descendants. For example, using our rooted tree from above, the subtree of h would be



M-ary Tree

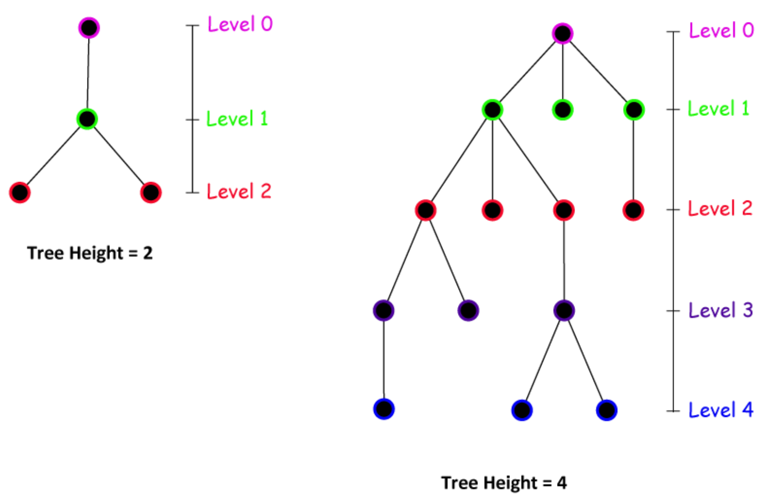
Additionally, a rooted tree is called an **m-ary tree** if every internal node has no more than m children. And a full m-ary tree occurs when every internal node has precisely m children. An m-ary tree with m=2 is called a binary tree. Below is an example of some m-ary trees.



Levels

Furthermore, a tree's vertices are organized into levels, based on how many edges or branches away from the root they are. The root is defined to be level 0, and its children are level 1, their children are level 2, and so forth. And the height of a tree is the maximum number of levels from root to leaf.

For example, let's determine the level and height for the following trees.



Theorems

Moreover, some essential theorems and properties help us determine the number of edges, vertices, leaves, level, and height.

- A tree with n vertices has $n-1$ **edges**
- A full m-ary tree with i internal vertices contains $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves
- A full m-ary tree with l leaves has $n = \frac{ml-1}{m-1}$ vertices and $i = \frac{l-1}{m-1}$ internal vertices
- A full m-ary tree with n vertices has $i = \frac{n-1}{m}$ internal vertices and $l = \frac{n(m-1)+1}{m}$ leaves
- A rooted m-ary tree of height h is balanced if all leaves are at level h or $h-1$
- There are at most m^h leaves in an m-ary tree with height h.
- If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$ and if the full m-ary tree is full and balanced then $h = \lceil \log_m l \rceil$

Example – Vertices

How many vertices does a full 8-ary tree with 121 internal vertices have?

$$n = mi + 1, \text{ where } m = 8 \text{ , and } i = 121$$
$$n = (8)(121) + 1$$
$$n = 969$$

Example – Leaves

Assume there is a 3-ary tree with 103 vertices. How many leaves does it have?

$$l = \frac{n(m-1)+1}{m}, \text{ where } m = 3, \text{ and } n = 103$$
$$l = \frac{103(3-1)+1}{3}$$
$$l = 69$$

Example – Internal Vertices

How many internal vertices does a full 5-ary tree with 101 leaves have?

$$i = \frac{l-1}{m-1}\text{, where } m=5\text{, and } l=101$$

$$i = \frac{(101-1)}{(5-1)}$$

$$i = 25$$

Example – Edges

Determine the number of edges a full 5-ary tree with 121 leaves have?

$$edges = n-1\text{, where } m=5,\ l=121\text{, and } n=\frac{ml-1}{m-1}$$

$$edges = \left(\frac{ml-1}{m-1}\right)-1$$

$$edges = \left(\frac{5\cdot 121-1}{5-1}\right)-1$$

$$edges = \left(\frac{604}{4}\right)-1$$

$$edges = 150$$