What is Tree?

- A Tree is a collection of nodes connected by edges or lines.
- The collection can be empty.
- If not empty a tree consists of a distinguished node R (the *root*), and zero or more nonempty *subtrees* T_1, T_2, ..., T_k, each of whose roots are connected by a directed *edge* from R.
- In nature, Trees have a very distinct feature compare to trees in data structure. Trees in nature grow upward while in data structure, it grows in the opposite way. Its root is on the top and the leaves are at the bottom.
- Unlike arrays, linked list, stacks and queues which are linear data structures, Trees are hierarchical data structure

Why Trees?

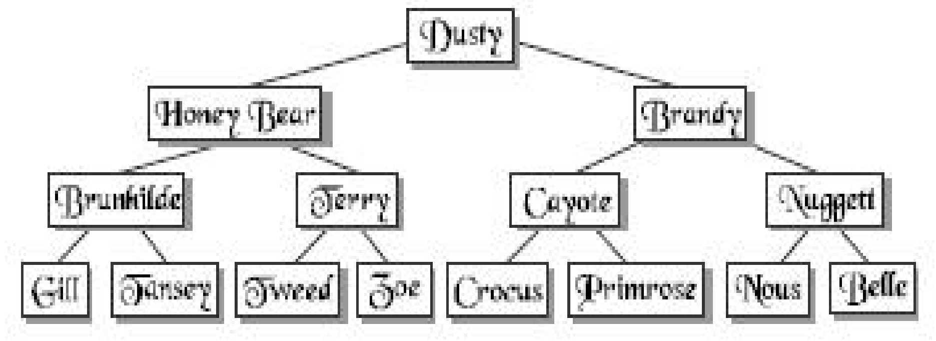
- One reason to use trees might be because you want to store information that naturally forms a hierarchy. For example, the file system on a computer.
- Trees (with some ordering e.g., BST) provide moderate access/search (quicker than Linked List and slower than arrays).
- Trees provide moderate insertion/deletion (quicker than Arrays and slower than Unordered Linked Lists).

Main Application of Trees

- Manipulate hierarchical data.
- Make information easy to search (see tree traversal).
- Manipulate sorted lists of data.
- As a workflow for compositing digital images for visual effects.
- Router algorithms
- Form of a multi-stage decision-making

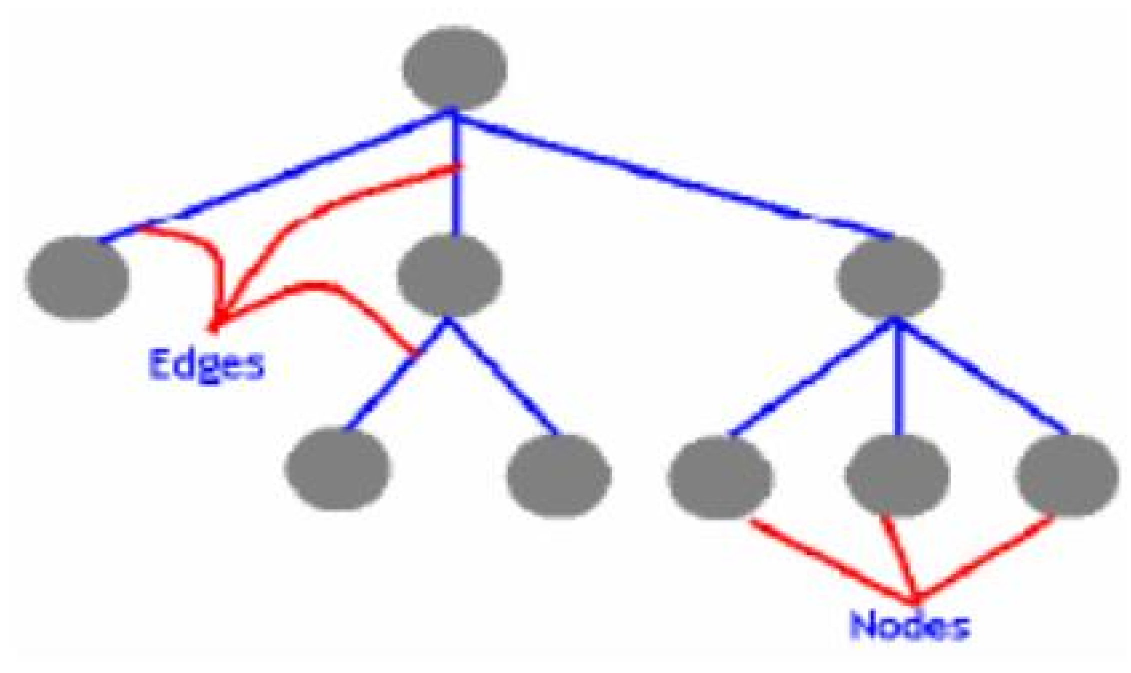
Operations Used In Trees

- Trees can perform quick insertion and deletion. Obvious examples of TREES are **genealogies** and **organizational charts.** Below is an example:



Trees – Data Structure

A TREE is composed of nodes connected by edges or lines



Basic Terminologies

- **Nodes** — these are what the trees are composed of. These are the ones being connected by edges or lines. These contain a value of represent a data structure.
- **Root Node** — this is a node at the topmost of the tree and where the tree commonly begins; definitely this node will have no parents at all. Any nodes can be a root node as long as it is rooted as a parent node. You can access this leaf node through the edges or lines.
- **Edges/Lines/Paths** — these are lines connecting to nodes that describe their relationship. These lines represent convenience in using a time because of easy access for the program; just by passing through the paths along the lines, you can access any nodes you want. Generally, you are allowed only once in accessing the nodes and that are from root downward.
- **Leaf Nodes** — these are nodes that are at the bottommost level of the tree and so, they have no any children. Note that these can only be one and root node but there can be many leaves. Leaves also known as terminal nodes have a zero degree.
- **Internal Nodes / Child Nodes** — These are the nodes that are inside the tree or the nodes below a given node. They are also known as SIBLING NODE.
- **Subtree** — a subtree is viewed as a complete tree itself which consist of children and its children's and so on. The subtree corresponding to the root node is called the entire tree; the subtree corresponding to any other node is called a proper subtree.
- **Visiting** — a node is said to be visited when a program control arrives at the node to perform operations such as checking the value of one of its data fields or simply to display it. Passing over a node on the path is not considered as visiting the node.

- **Traversing** — to visit all the nodes in a specific order is called traverse or more commonly known as walking the tree and the action is walk.
- **Levels** — the levels are generations that a tree has starting on the root.
- **Degree** — this refers to the number of nodes in a subtree.
- **Depth Of The Line** — the depth of the tree is the highest level.
- **Keys** — keys in tree diagrams represent the value inside a node represented in a circle.

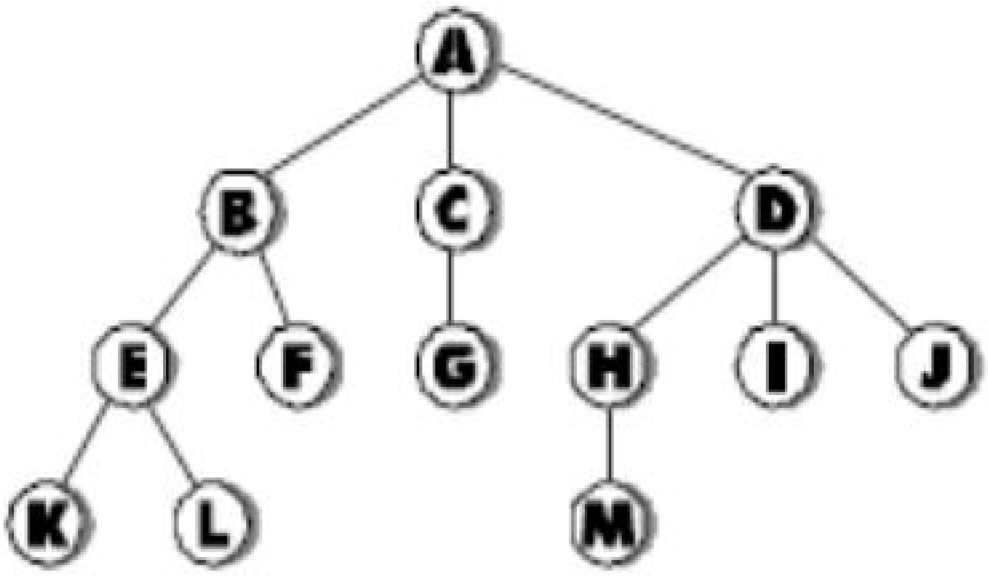


Figure 2.

TERMINOLOGIES	ANSWER
Root Node	A
Internal Nodes	B, C, D, E, H,

Leaves or Leaf Nodes	K, L, M, F, G, I, J
Levels	There are 4 levels in a tree: • Level 1 – A • Level 2 – B, C, D • Level 3 – ? • Level 4 – ?
Subtree	• B, E, F • C, G • ?
Depth of the Tree	4
Degree	• Degree of A is 3 • Degree of B is 2 • Degree of C is 1 • Degree of K is 0 • The maximum degree of the tree is 3