TREE

- **Tree**: - A tree is a finite non-empty set of elements (nodes) linked together to simulate a hierarchy. It is a non- linear data structure.

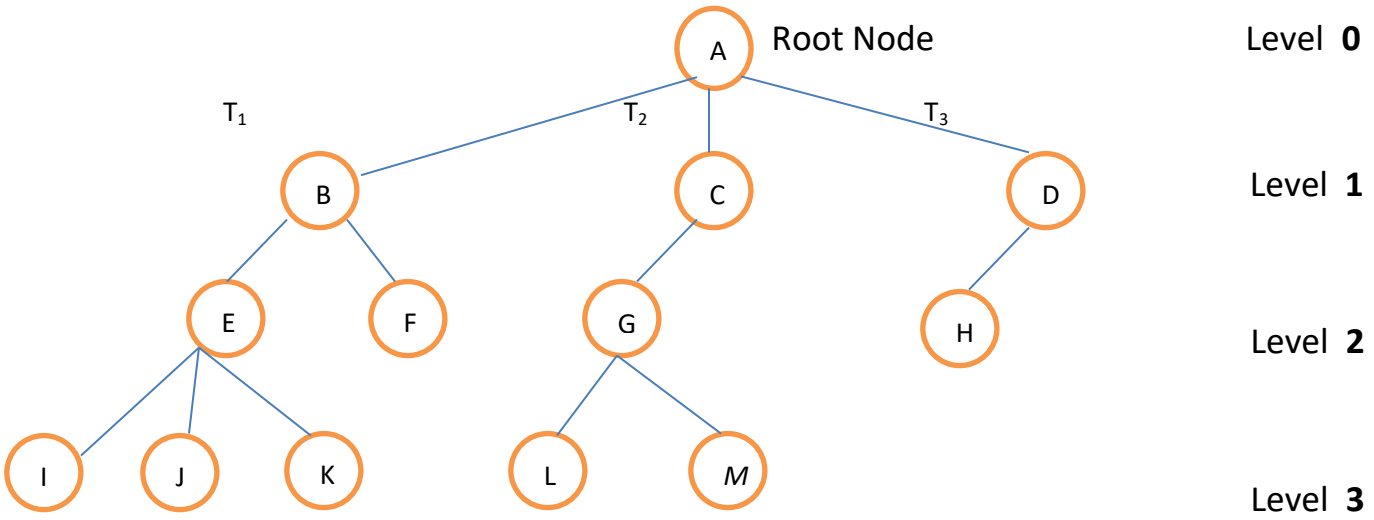

Figure of a TREE

- **Root**: - The root node is the topmost node in the tree. If root node **R= NULL**, then it means the tree is empty. In the above tree, **A** is the root node.
- **Sub-trees**: - If the root node **R** is not **NULL**, then the trees $\mathbf{T_1}$,$\mathbf{T_2}$ and $\mathbf{T_3}$ are called the sub-trees of **R**.
- **Node**: - Each data item in a tree is called a node. There are **13** nodes in the above tree.
- **Degree of a node**: - It is the number of sub trees of a node in a given tree. In the above tree:
    - The degree of node **A** is **3**.
    - The degree of node **B** is **2**.
    - **The degree of node H is 0.**
- **Degree of a tree**: - It is the maximum degree of nodes among all nodes in a given tree. In the above tree the node **A** node **E** have degree **3** that is maximum. So, the degree of the above tree is **3**.
- **Parent:** - Immediate predecessor of a node is called its parent. Example: - **G** is parent of **L** and **M**.

- **<u>Child</u>**: - Immediate successor(s) of a node is called its child/children. Example: - **L** and **M** are children of **G**.
- **<u>Leaf node/Terminal node</u>**: - A node with degree **0 (zero)** is called a leaf node or terminal node.

  OR

  A node that has no children is called the leaf node or the terminal node. Example: - I, J, K, F, L, M and H are leaf nodes of the above tree.
- **<u>Non-terminal node(s)</u>**: - Any node (except the **root node**) whose degree is not **zero (0)** is called non-terminal node. Non-terminal nodes are the intermediate nodes in traversing the given tree from its root node to the terminal nodes (leaves). There are **5** non- terminal nodes (B, C, D, E, G).
- **<u>Edge</u>**: - It is the connecting line of two nodes. That is, the line drawn from one node to another node is called an edge. Example: - (A,B)
- **<u>Path</u>**: - It is a sequence of consecutive edges from the source node to the destination node. In the above tree, the path between **A** and **J** is given by the node pairs- (A,B), (B,E) and(E,J).
- **<u>Ancestor node</u>**: - Any predecessor node on the path from root to that node is called an Ancestor node. The root node does not have any ancestors. Example: - Nodes A, C and G are the ancestors of node L.
- **<u>Descendant node</u>**: - Any successor node on any path from the node to a leaf node. Leaf nodes do not have any descendants. Example: - Nodes **C, G, L** and **M** are the descendants of node **A**.
- **<u>Siblings</u>**: - The children nodes of a given parent node are called siblings. They are also called brothers. Example: - **E** and **F** are siblings of parent node **B**.
- **<u>Depth of a node</u>**: - Length of path from root to that node or number of edges from root to that node. Example: - **F= 2, A= 0**.
- **<u>Height of a node</u>**: - Number of edges in the longest path from that node to a leaf node. Example: - **B= 2**.
- **<u>Height of a tree</u>**: - Number of edges in the longest path from root node to a leaf node. Example: - **Height of the above tree= 3**.

- **<u>Level of a node</u>**: - Distance of root to the given node or number of edges from root to the given node. Example: - **G= 2**.

  <u>NOTE</u>: -

  - Always level of node = Depth of node.
  - Level of a tree = Height of that tree.
  - Level of a node ≠ Height of that node.
  - If a tree has '**n**' nodes than it should have exactly **(n-1)** edges.