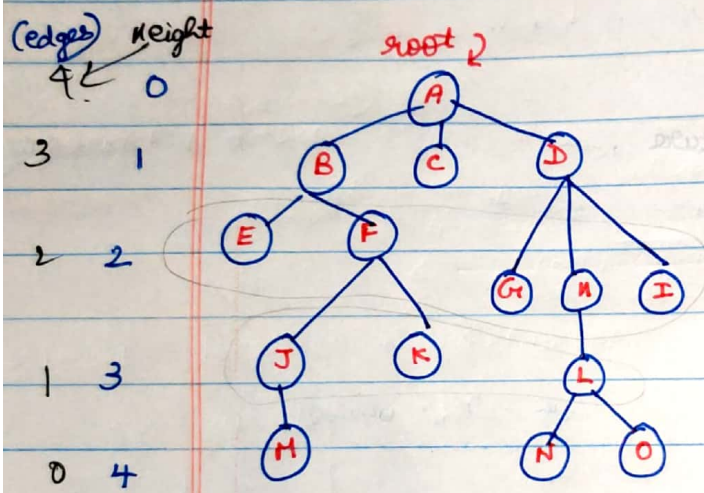
Trees

* Trees are the collection of vertices & edges
* If there are n nodes then $(n-1)$ edges

since all the nodes have edges
but parent node doesn't have edge.

* Trees are the collection of nodes, where one of the node is root node & the rest of the nodes are divided into disjoint subsets and each subset is a tree / sub-tree.



1) Root :

2) Parent :

3) Child : E, F are the childs of B.

G, H, I are the children of D.

4) Siblings : Siblings are the children of same parent.

Eg: E, F and G, H, I are siblings.

J, K and N, O

5) **Descendants**: Descendants are the set of nodes which can be reached from a particular node / under that node.

Eg: For B / E, F, J, K, M are all the descendants of B.

For D / G, H, I, L, N, O are all the descendants of D.

For L / N, O are descendants of L.

(ie) All the nodes that are reachable from D are the descends of D.

6) **Ancestors**:

Let a Node be M. From M the nodes which are taken to reach the root Node, are the ~~descendors~~ ancestors of A

Eg: For M; J, F, B, A are ancestors

For N; L, M, D, A are ancestors

7) **Degree**

Degree of a node is the number of children, it is having.

↓

Should be direct children

Eg: Degree of D is 3

L is 2

A is 3

N, K, O is 0

8) Internal / External nodes → Nodes with degree 0
Eg: E, H, G, C, I, N, O

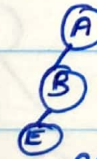
External nodes ↓
leaf nodes

i) non-leaf nodes

ii) Non-terminal nodes ~~Non~~ terminal nodes

Nodes with degree > 0 Eg: F, J, K, L

9) Level: Level is the path which is taken by number of nodes
Eg: Level 3 ⇒ A, D, I
Level 3 ⇒ A, B, F (or) A, B, E

10) Height of a tree:
Height is the path which is taken by number of edges
Eg:


For E, height is 2, since the number of edges from root to E is 2

11) Forest: Forest is a collection of trees.
Eg: Remove A