جامـــــعة المـــــستـقبـل
AL MUSTAQBAL UNIVERSITY

قســـم الامـــــن الـــــــــسيبرانـــــــي

Department of Cyber Security

Subject:

Searching And Sorting Algorithms

Class:

Second

Lecturer:

Asst. Prof. Dr. Ali Kadhum Al-Quraby

Lecture: (3)

Introduction to Graph

1.1 Introduction to Graph

Graph is a nonlinear data structure, it contains a set of points known as nodes (or vertices) and set of links known as edges (or Arcs) which connects the vertices.

A graph is defined as follows:

Graph is a collection of vertices and arcs which connects vertices in the graph.

Graph is a collection of nodes and edges which connects nodes in the graph.

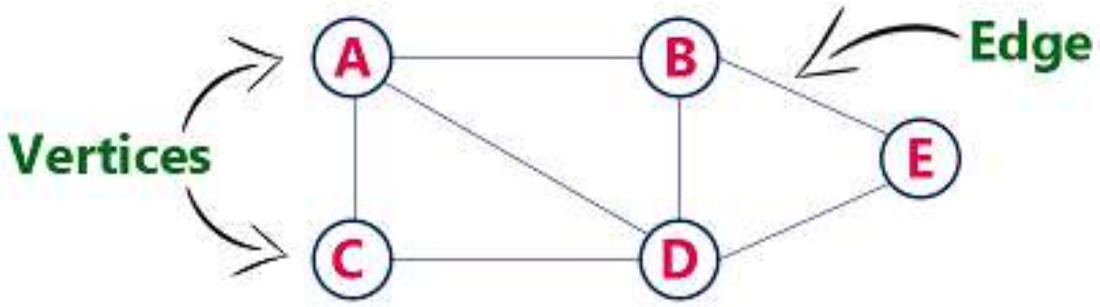
Generally, a graph G is represented as G = (V , E), where V is set of vertices and E is set of edges.

Example

The following is a graph with 5 vertices and 6 edges.

This graph G can be defined as G = (V , E)

Where V = {A,B,C,D,E} and E = {(A,B),(A,C)(A,D),(B,D),(C,D),(B,E),(D, E)}.



1.1.1 Graph Terminology

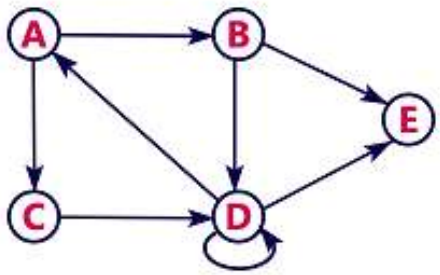
We use the following terms in graph data structure...

Vertex: A individual data element of a graph is called as Vertex. Vertex is also known as node. In above example graph, A, B, C, D & E are known as vertices.

Edge: An edge is a connecting link between two vertices. Edge is also known as Arc. An edge is represented as (startingVertex, endingVertex). For example, in above graph, the link between vertices A and B is represented as (A,B). In above example graph, there are 7 edges (i.e., (A,B), (A,C), (A,D), (B,D), (B,E), (C,D), (D,E)).

Undirected Graph: A graph with only undirected edges is said to be undirected graph as in the above.

Directed Graph: A graph with only directed edges is said to be directed graph as in the figure below:



Connected graph: A graph G is called connected if every two of its vertices are connected.

Disconnected graph: A graph that is called not connected if some of its vertices is disconnected.

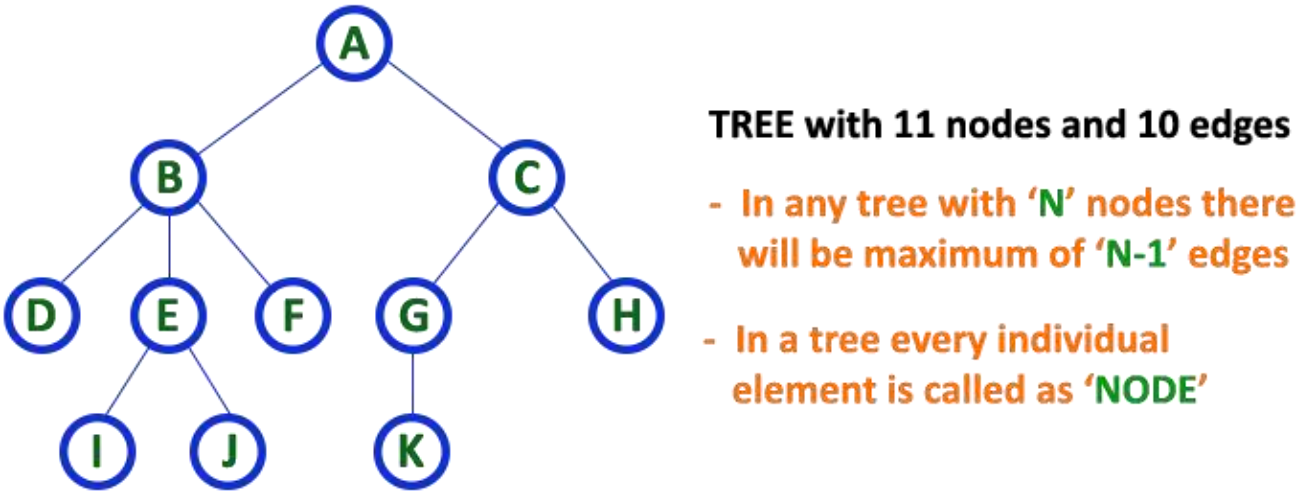
1.2 Trees

A tree data structure can be defined as follows...

A **connected acyclic graph** is called a tree. In other words, tree is a connected graph with no cycles .

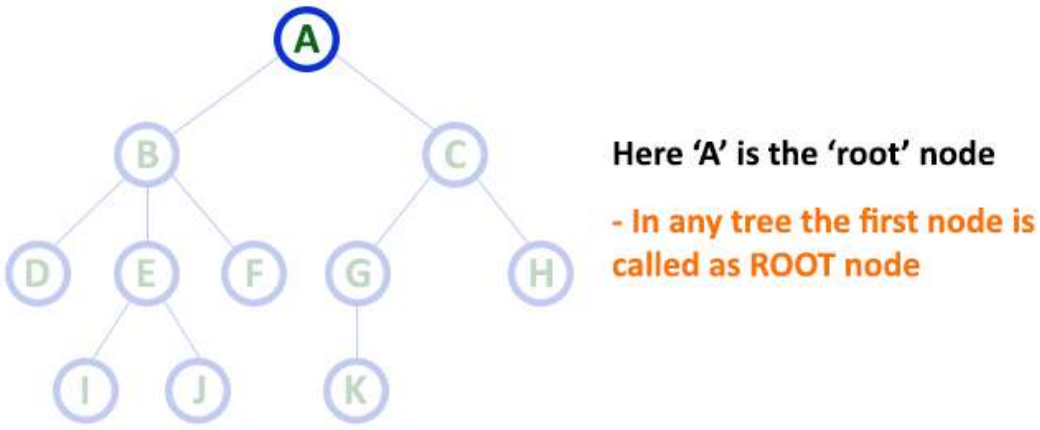
In a tree data structure, if we have **N** number of nodes then we can have a maximum of **N-1** number of links.

Example

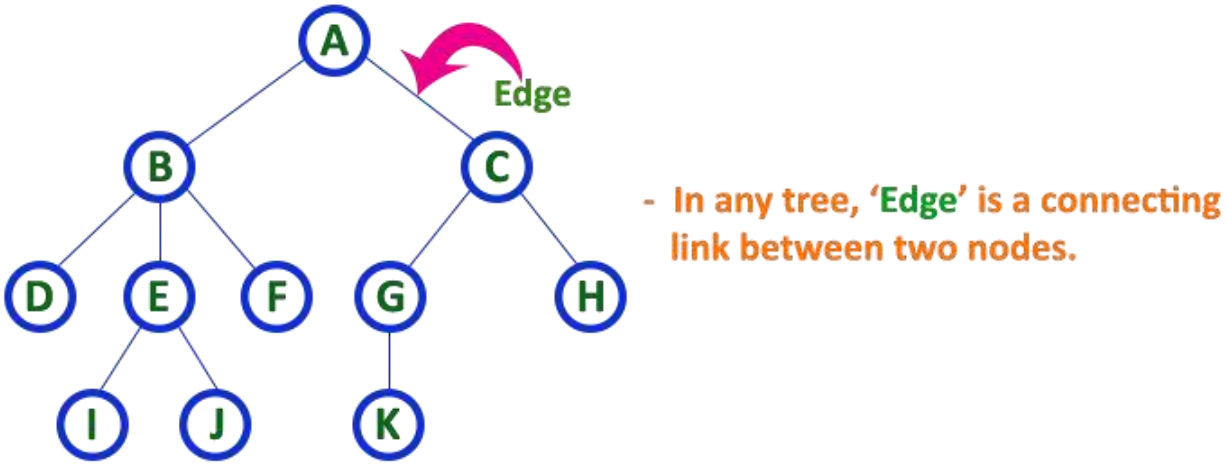


In a tree data structure, we use the following terminology...

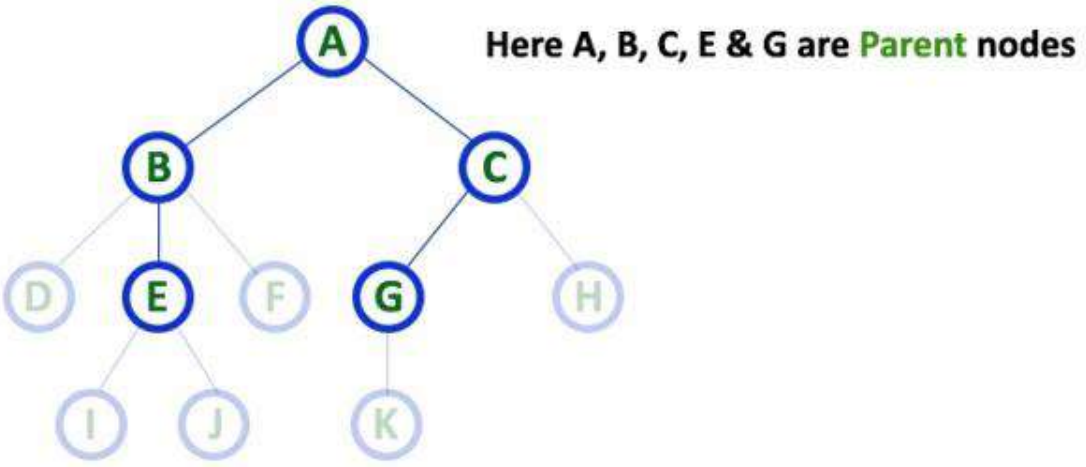
1. Root : In a tree data structure, the first node is called as **Root Node**. Every tree must have root node. We can say that root node is the origin of tree data structure. In any tree, there must be only one root node.



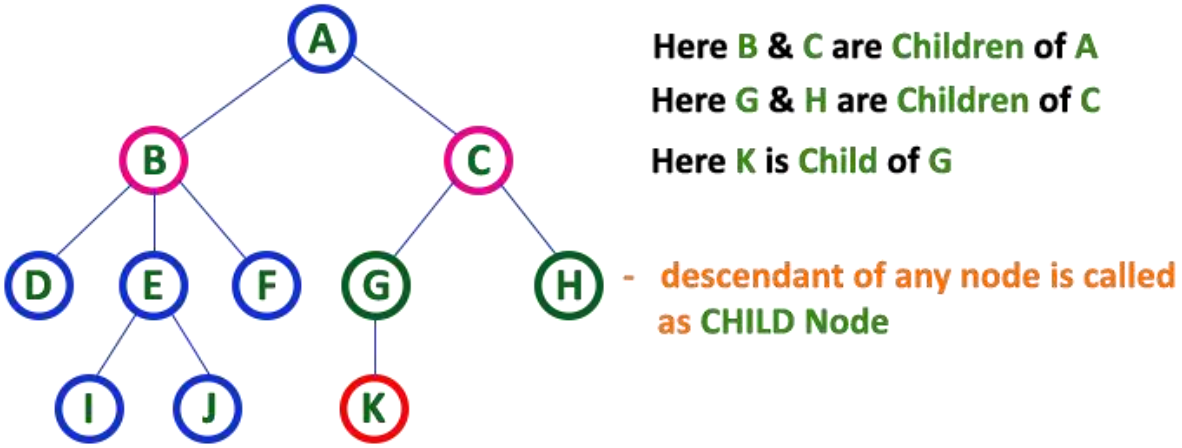
2. Edge: In a tree data structure, the connecting link between any two nodes is called as **EDGE**. In a tree with '**N**' number of nodes there will be a maximum of '**N-1**' number of edges.



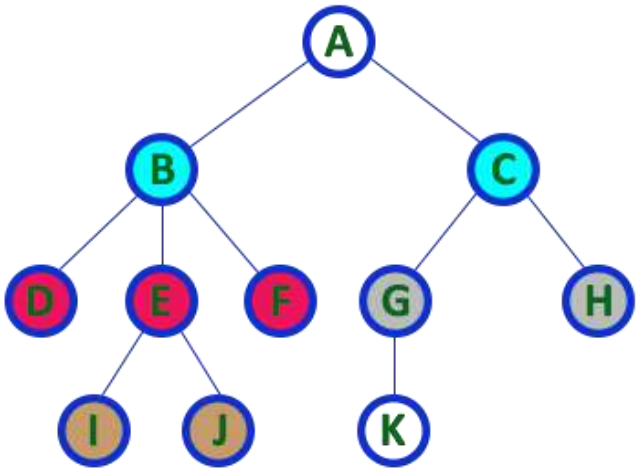
3. Parent: In a tree data structure, the node which is predecessor of any node is called as **PARENT NODE**. In simple words, the node which has branch from it to any other node is called as parent node. Parent node can also be defined as "**The node which has child / children**".



4. Child: In a tree data structure, the node which is descendant of any node is called as **CHILD Node**. In simple words, the node which has a link from its parent node is called as child node. In a tree, any parent node can have any number of child nodes. In a tree, all the nodes except root are child nodes.



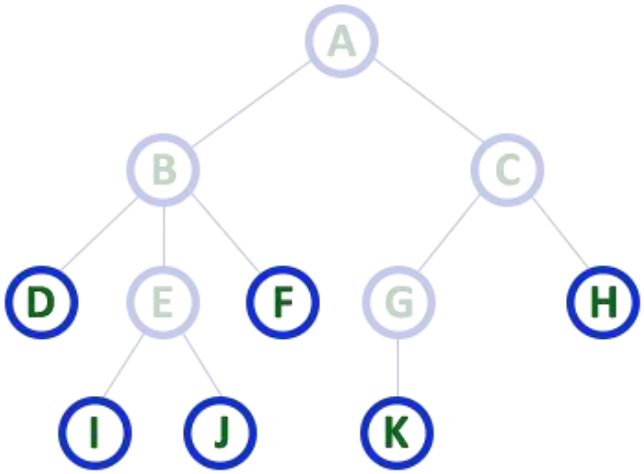
5. Siblings: In a tree data structure, nodes which belong to same Parent are called as **SIBLINGS**. In simple words, the nodes with same parent are called as Sibling nodes.



Here B & C are Siblings
Here D E & F are Siblings
Here G & H are Siblings
Here I & J are Siblings

- In any tree the nodes which has same Parent are called 'Siblings'
- The children of a Parent are called 'Siblings'

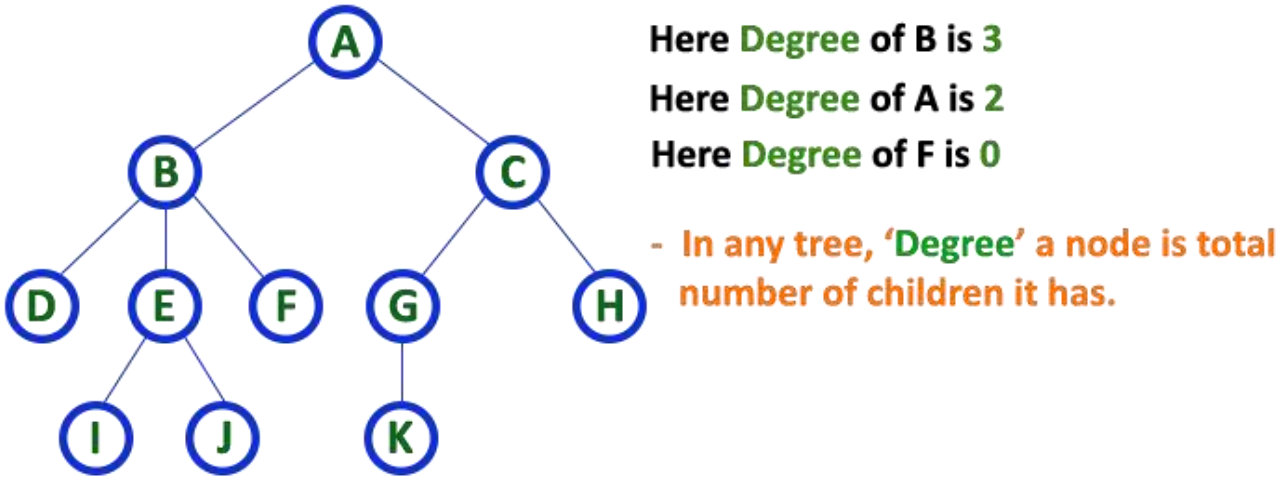
6. Leaf: In a tree data structure, the node which does not have a child is called as **LEAF Node**. In simple words, a leaf is a node with no child.
leaf node is also called as '**Terminal**' node.



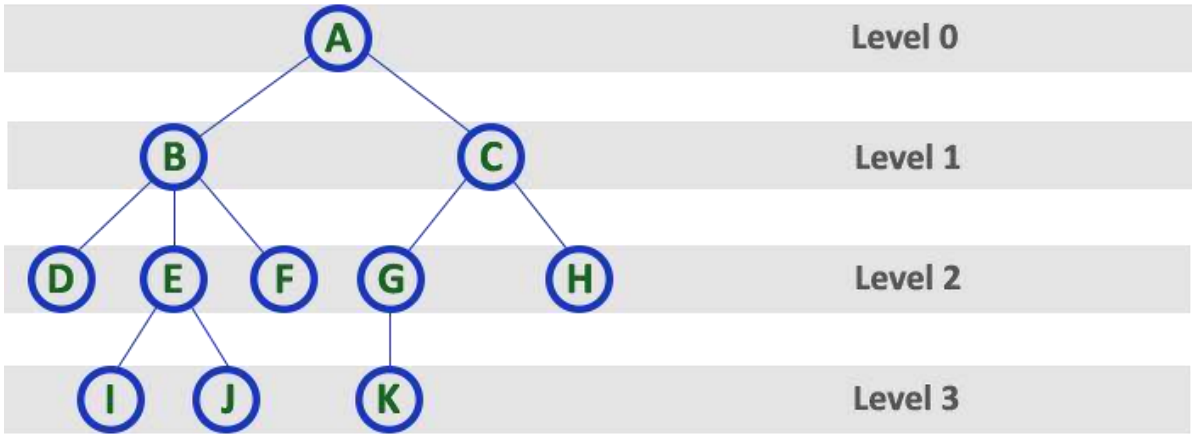
Here D, I, J, F, K & H are Leaf nodes

- In any tree the node which does not have children is called 'Leaf'
- A node without successors is called a 'leaf' node

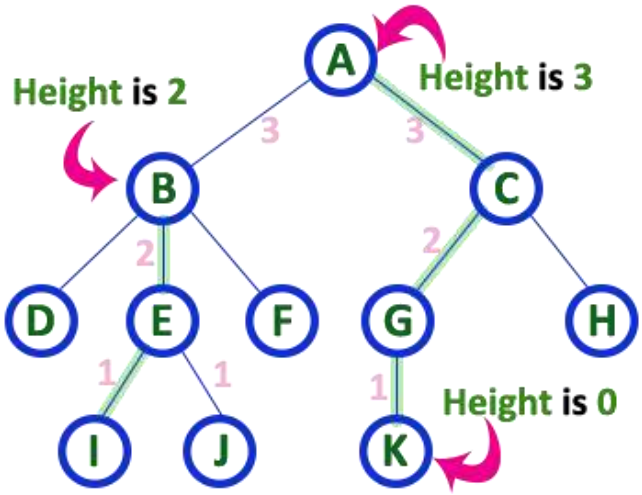
7. Degree: In a tree data structure, the total number of children of a node is called as **DEGREE** of that Node. In simple words, the Degree of a node is total number of children it has. The highest degree of a node among all the nodes in a tree is called as '**Degree of Tree**'



8. Level: In a tree data structure, the root node is said to be at Level 0 and the children of root node are at Level 1 and the children of the nodes which are at Level 1 will be at Level 2 and so on... In simple words, in a tree each step from top to bottom is called as a Level and the Level count starts with '0' and incremented by one at each level (Step).



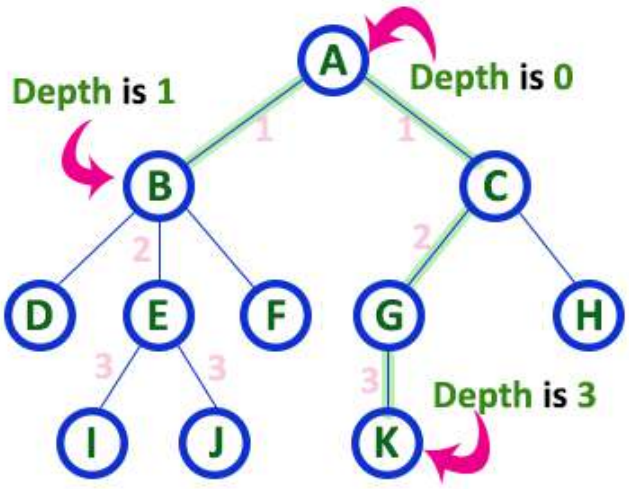
9. Height: In a tree data structure, the total number of edges from leaf node to a particular node in the longest path is called as **HEIGHT** of that Node. In a tree, height of the root node is said to be **height of the tree**. In a tree, **height of all leaf nodes is '0'.**



Here Height of tree is 3

- In any tree, 'Height of Node' is total number of Edges from leaf to that node in longest path.
- In any tree, 'Height of Tree' is the height of the root node.

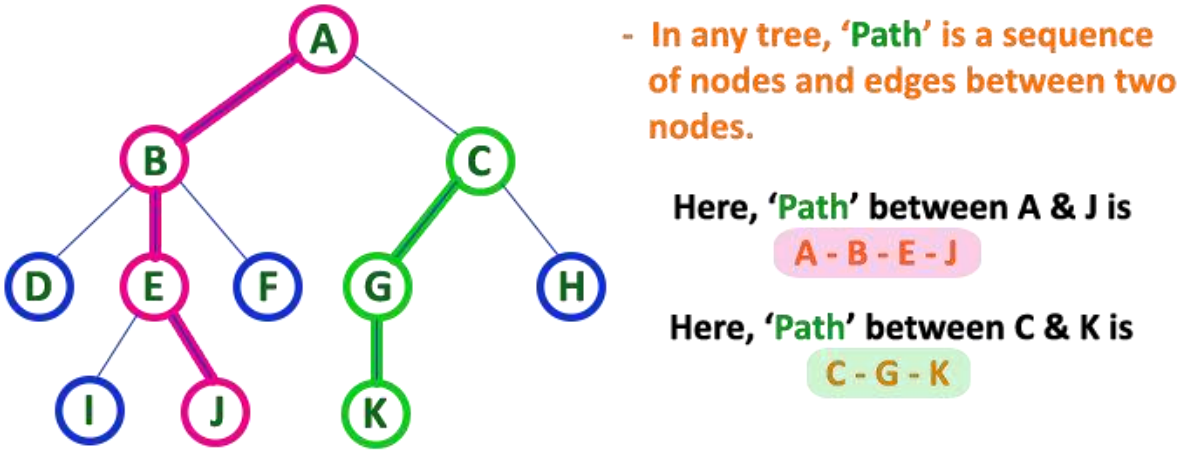
10. Depth: In a tree data structure, the total number of edges from root node to a particular node is called as **DEPTH** of that Node. In a tree, the total number of edges from root node to a leaf node in the longest path is said to be **Depth of the tree**. In simple words, the highest depth of any leaf node in a tree is said to be depth of that tree. In a tree, **depth of the root node is '0'.**



Here Depth of tree is 3

- In any tree, 'Depth of Node' is total number of Edges from root to that node.
- In any tree, 'Depth of Tree' is total number of edges from root to leaf in the longest path.

11. Path: In a tree data structure, the sequence of Nodes and Edges from one node to another node is called as **PATH** between that two Nodes. **<u>Length of a Path</u>** is total number of nodes in that path. In below example **the path A - B - E - J has length4.**



12. Sub Tree

In a tree data structure, each child from a node forms a subtree recursively. Every child node will form a subtree on its parent node.