**Concept Application & Algorithmic Part**

**Question 1: (20 points)**

Suppose we have the following Binary Search Tree (BST).

Shape

Description automatically generated

**Algorithm Write up**. Write an algorithm to determine the nodes at k given distance from root.

**Example -**  
Nodes at distance 2 from root=> 2, 6, 10, 14

Nodes at distance 1 from root=> 4,12   
Nodes at distance 3 from root=> 1, 3, 5, 7, 9, 11, 13, 15

Algorithm:

* Invoke recursive method
* Input )level, root(
* Define base case (Termination Condition)

If (root == null) { return }

* If (level == 0) { Print root.number }

Else

Repeating the method by its generic formula

{ Print (root.left, level-1)

Print (root.right, level -1) }

**Question 2: (20 points)**

public BSTnode parent(BSTnode p) {

return parent(root, p);

}

private BSTnode parent(BSTnode root, BSTnode p) {

if (root == null || root == p)

return null;

if (root.getLeft()==p || root.getRight()==p)

return root;

if (p.getData()< root.getData())

return parent(root.getLeft(), p);

else if (p.getData() >root.getData())

return parent(root.getRight(), p);

else return null;

|  |
| --- |
| Using the same BST tree in Question 1, show the complete tracing of **Parent()** method (given in the code of BST slides) to find the parent of the node that stores the value **13**. You are required to show the complete tracing (drawing or boxes) of this method. |

4

14

14

14

Parent(14,13);

Return 14;

Parent(8,13);

Parent(12,13);