4-Trees

Practical exercises
Question 1. Write a Java program to implement a binary search tree of integer values with the following operations:

1. boolean isEmpty() - return true if a tree is empty, return false otherwise.
2. void clear() - clear a tree.
3. Node search(int x) - Search a node having value x. Return a reference to that node if found, return null otherwise.
4. void insert(int x) - check if the key x does not exists in a tree then insert new node with value x into the tree.
5. void breadth() - traverse a tree by breadth first search/level order.
6. void preorder(Node p) - recursive preorder traverse of a tree.
7. void inorder(Node p) - recursive inorder traverse of a tree.
8. void postorder(Node p) - recursive postorder traverse of a tree.
9. int count() - count and return number of nodes in the tree.
10. Node min() - find and return the node with minimum value in the tree.
11. Node max() - find and return the node with maximum value in the tree.
12. int sum() - return the sum of all values in the tree.
13. int avg() - return the average of all values in the tree.
14. The height of a tree is the maximum number of edges on a path from the root to a leaf node (thus the height of a tree with root only is 0). Write a function that returns the height of a binary tree.

Question 2: implement tree sort

Social Constructivism: void dele(int x) - delete a node having value x.