1. (7 marks) Consider the following binary tree:

```
     q
    / \
   m   t
  /   / \
 d   k   j
 /     / \
 b     p   y
      /   \
     n     w
```

a) (2 marks) Is this tree a binary search tree? Explain your answer.

b) (1 mark) List the ancestor(s) of node k.

c) (1 mark) List the descendant(s) of node m.

d) (1 mark) What is the depth of the tree?

e) (2 marks) Give the order in which the nodes are visited by a preorder traversal.
2. **(4 marks)** Draw a diagram of the binary search tree created by adding the following values in the order given to an initially empty tree:

```
26 10 30 27 8 19 18 21 25 6
```

3. **(9 marks)** Write a class `BTNode` Java method

```java
public static double subTInv(BTNode root)
```

which, when called with a reference to the root of a binary tree, returns the quantity $1.0/(WL \times 1.0)$, where $WL$ is the number of nodes in the tree with null left subtrees. For example, the value returned for the tree in Question 1 is $1.0/6.0 = 0.16666\ldots$

**Hint:** You may find it useful to break this into two methods, one of which does a recursive traversal of the tree.