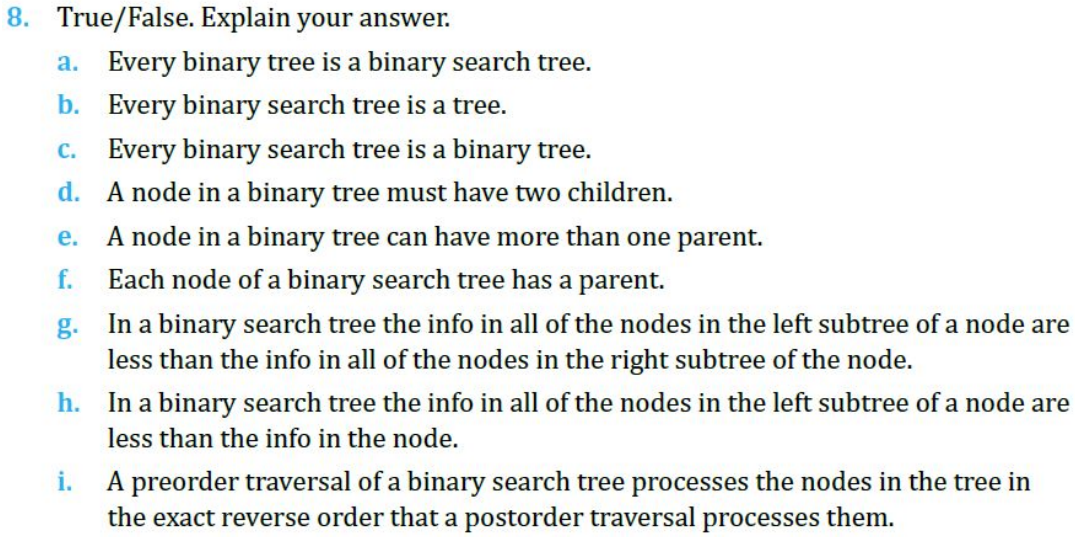
Short Answers

P1. (3pts) What is the purpose of using the header and trailer nodes in a linked list?
These two nodes serve to merely simplify the insertion and deletion algorithms and are not part of the actual list. The actual list is between these two nodes so the header will be less than the least possible element and the tail will be the larger than the largest possible element.

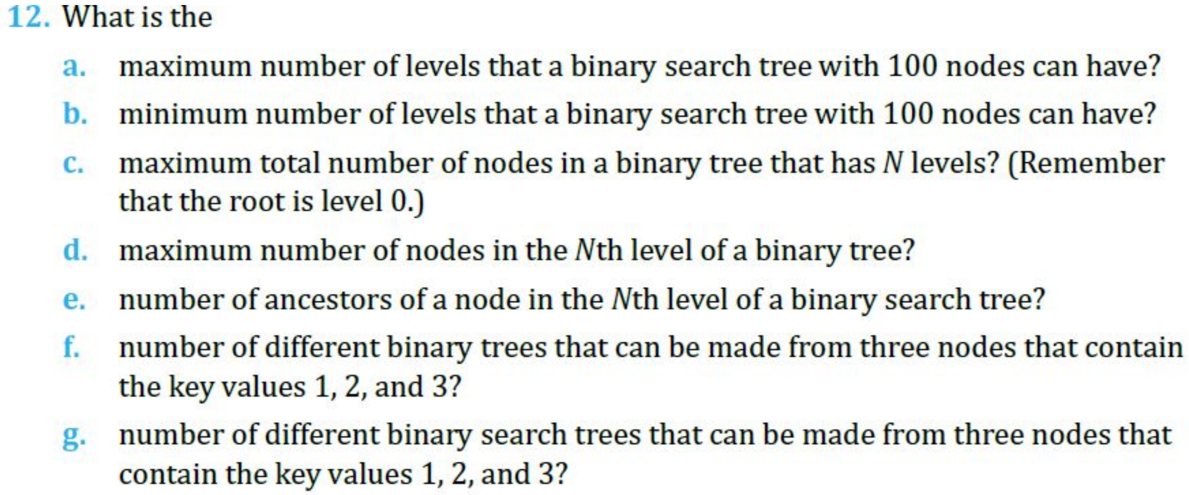
P2. (9pts, 1pt each) Q8 on page 489



8. True/False. Explain your answer.
   a. Every binary tree is a binary search tree.
   b. Every binary search tree is a tree.
   c. Every binary search tree is a binary tree.
   d. A node in a binary tree must have two children.
   e. A node in a binary tree can have more than one parent.
   f. Each node of a binary search tree has a parent.
   g. In a binary search tree the info in all of the nodes in the left subtree of a node are less than the info in all of the nodes in the right subtree of the node.
   h. In a binary search tree the info in all of the nodes in the left subtree of a node are less than the info in the node.
   i. A preorder traversal of a binary search tree processes the nodes in the tree in the exact reverse order that a postorder traversal processes them.

a) False, a binary tree is not ordered like a binary search tree
b) True, the hierarchy of trees goes down following: tree < binary tree < binary search tree
c) True, Since binary search trees are higher on the scale they include binary trees/trees
d) False, there can be nodes with leaf nodes(only one child) or no children at all.
e) False, all nodes in a BST must have 1 parent to keep the structure.
f) False, root node has no parent
g) True, since the BST structure dictates putting #'s < the root to the left this is valid.
h) True, if we are speaking of the root node yes because if they were not smaller they would be on the right side.
i) False, Pre-order traversal: root -> left subtree -> right subtree
   Post-order traversal: left subtree -> right subtree -> root

P3. (7pts, 1pt each) Q12 on page 489



12. What is the

a. maximum number of levels that a binary search tree with 100 nodes can have?

b. minimum number of levels that a binary search tree with 100 nodes can have?

c. maximum total number of nodes in a binary tree that has *N* levels? (Remember that the root is level 0.)

d. maximum number of nodes in the *N*th level of a binary tree?

e. number of ancestors of a node in the *N*th level of a binary search tree?

f. number of different binary trees that can be made from three nodes that contain the key values 1, 2, and 3?

g. number of different binary search trees that can be made from three nodes that contain the key values 1, 2, and 3?

a) 100
b) 7
c) 2^(n)-1
d) 2^(n-1)
e) n-1
f) 30; n = 3; (n!)*(2n)!/(n+1)!n! (non-structural)
g) 5; n = 3; (2n)!/(n+1)!n!

P4. (10pts) Draw a binary search tree whose element are inserted in the following order:
50 72 96 94 107 26 12 9 15 10

```
              50
            /    \
         26       72
        /  \      /  \
      12           96
     /  \          /  \
   9   15      94   107
  / \   / \     / \    / \
   10
```