Read Ch. 6 Section 3 (pp. 402+) closely and answer the following questions.

- 1. What does the author mean by a binary search tree (BST) having a "full shape" on p. 405?
- 2. Give a mathematical expression for the maximum number of nodes in one branch of a BST with a full shape.
- 3. What is the big O search efficiency for a full BST?
- 4. What are the 3 cases of node deletion on a binary search tree? Draw a diagram to illustrate each case.

5. Why is the loop

```
while (q->left != 0)
{
    q = q->right;
    qParent = qParent->right;
}
```

important in the code segment at the bottom of p. 409?

6. How does the binary search tree approach to organizing sorted data allow us to locate data much faster than a linked list representation?

7. FULLY explain the memory consumption of the binary search tree approach to keeping data organized and sorted.

- 8. What is the worst case big O search efficiency for a BST?
- 9. Given the advantages and disadvantages of using of binary search trees and linked lists to sorting data, when should a programmer use a BST rather than a linked list? Continue your answer onto the back of the paper if necessary.