Problem 3:

This solution achieves O(log N) time because it is a binary search tree where the key that is being searched for has a very exact direction that its going towards. According to the binary search tree if the key is greater than the parent it will continue searching the right side of the tree. If the key is smaller than the parent it will continue searching the left side of the tree.

**public** Node search (Node key){

**while** (key != **null**){

**if** (key > parent){

search(node.right);

}

**else**(key < parent){

search (node.left);

}

}

}