ECE 250 *Algorithms and Data Structure*
Department of Electrical and Computer Engineering
University of Waterloo
Please send any comments or criticisms to dwharder@alumni.uwaterloo.ca
with the subject ECE 250 Questions 6.1.
Assistances and comments will be acknowledged.

6.1*b* For a perfect binary tree, the average depth is

$$\frac{\sum_{h=0}^{3} h2^h}{15} = \frac{34}{15} = 2.2666\cdots.$$

For this tree, it is

$$\frac{1\cdot 0 + 2\cdot 1 + 4\cdot 2 + 3\cdot 3 + 2\cdot 4 + 2\cdot 5 + 1\cdot 6}{15} = \frac{43}{15} = 2.8666\cdots.$$

6.1*c* Any numbers in 51, 52, …, 56 and 58.

6.1*d* The resulting tree would have 50 in the root and the left sub-tree of its right-sub-tree would be the linked list containing 62, 59 and 57.

6.1*e* For example, updates to the insert member function could be:

```
template <typename Type>
bool Binary_search_node<Type>::insert( Type const &obj,
                                       Binary_search_node *&ptr_to_this  ) {
    if ( empty() ) {
       // the constructor sets tree_size to 1
       ptr_to_this = new Binary_search_node<Type>( obj );
       return true;
    } else if ( obj < retrieve() ) {
        if ( left()->insert( obj, left_tree ) ) {
            ++tree_size;
            return true;
        } else {
            return false;
        }
    } else if ( obj > retrieve() ) {
        if ( right()->insert( obj, right_tree ) ) {
            ++tree_size;
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
```

6.1*f* O(*n*)