Objective: To gain experience in implementing abstract data types (ADT’s).

Goal. To implement the dynamic dictionary ADT as an object template in C++. To gain experience in programming binary trees.

The (dynamic dictionary) abstract data type is specified as follows:

```
AbstractDataType dynamicDictionary{
  Instances: finite collections of zero or more ordered pairs of type (keyType, dataType)
  Operations:
    Create( ): Create an empty dynamic dictionary
    Destroy( ): Erase a dynamic dictionary
    Size( ): Return the number of ordered pairs stored in the dynamic dictionary
    IsEmpty( ): Return true if the dynamic dictionary is empty; false otherwise
    Insert(k,d): Insert the ordered pair (k,d) into the dynamic dictionary
    Remove(k ): Delete all ordered pairs (k,d) in the dynamic dictionary
    find(k ): Return the data value d if the ordered pair (k,d) is in the dynamic dictionary
}
```

You are to implement this ADT in the C++ programming language using the binary search tree as the underlying data structure. Credit will be awarded for efficient implementation. Extensively test your ADT, using the black-box testing technique. As with all programming assignments, you must submit a cover sheet, design plan, etc.