The Assignment:

Implement the bag template class from Section 10.5, using a binary search tree to store the items.

Purposes:

Ensure that you understand and can use binary search tree.

Before Starting:

Read all of Chapter 10, especially Sections 10.3 and 10.5.

Files that you must write:

1. bag6.h:
2. Header file for this version of the bag class. You don't have to write much of this file.
3. bag6.tpp.h:
4. The implementation file for the new bag class. I have written much of this to get you started.
   There are four functions in this implementation file that you must implement. These files are marked with the words STUDENT WORK.

Other files that you may find helpful:

1. bintree.h
2. : and bintree.tpp.h
3. This is the binary tree node template class from Section 10.3.
4. bagtest.cxx
5. : A simple interactive test program.
6. bagexam.cxx
7. : A non-interactive test program that will be used to grade the correctness of your bag class.

The Bag Class Using a Binary Search Tree
Discussion of the Assignment

Start by understanding the entire pseudocode for the binary search tree operations (from Section 10.5). Then read through the portions that I have already implemented for you. Implement the rest of your work in two parts: (1) The insert and count functions, and (2) The bst_remove_all and bst_remove_max functions. Don't move to step 2 until you have completely finished and tested step 1.

Since this is a template class, debugging can be more difficult (some debuggers don't permit breakpoints in a template function. To help in debugging, you can call b.debug() in a program to print the binary search tree for the bag b (using the format shown on page 505-506).

The bag and bintree templates are never compiled on their own, but in order to create bagexam.o, all the template files must be present in the current directory.

Please upload to canvas all the files including the bag.tpp.h file and the output file (as a .txt file).