CSCI 230, Spring 2015 **Due: Friday, February 27, 5pm**
Homework 5

Add three new remove methods to the BinarySearchTree class, page 116.

In the current **remove** method, which is to remain a method in the class, when a node with two children is removed, the node is not actually removed, but the data field of that node is replaced by the data in the of the smallest node in its right subtree, then the replacement node is recursively removed. (This is always the min element in the right subtree.) See the Figure 4.24, page 119.

All the methods will differ only in how they remove nodes with two children. The methods to be added are:

- **removeL** – which, on a call to remove a node with two children, replaces the data field of that node, with the largest element in its left subtree, and recursively remove that node. (This is the opposite of what the current method does, but works just as well.)
- **removeAlternating** – which alternately replaces with the largest in the left subtree and the smallest in the right subtree. (Note – this method can use remove and removeL.)
- **removeRandomly** – which will replace with either the largest element in the left subtree or the smallest element in the right subtree, making the choice randomly. (Note – this method can use remove and removeL.)

Your new methods are public and similar to the existing public **remove** method, in that they accepts a single parameter and return no value: **void remove(AnyType x)**.
Like **remove**, **removeL** should also call a private recursive method **BinaryNode<AnyType> removeL(AnyType x, BinaryNode<AnyType>).**

Upload the entire BST class, with the three additional methods.

Collaboration policy:
· Work alone.
· Ask questions of instructor or Santoshi Nitya.
· You **may** post to piazza questions/comments about the binary search code provided by the author, but you **may not** post questions or comments about your solution to this problem.
· No internet searching for solutions to the problem.
· You may, of course, use the internet for help with java coding, just not this problem.