Objective: To understand the implementation and performance of AVL trees, including their `add` and `find` methods.

To start the lab: Download and unzip the file at: www.cs.uni.edu/~fienup/cs1520s12/homework/hw6.zip

Step 1: Complete the `add` method code including `rotateRight` in the `hw6/avl.py` file. You will want to use your lecture 19 handout from class or print a new copy from:
http://www.cs.uni.edu/~fienup/cs1520s12/lectures/lec19_questions.pdf

You can test your AVL `add` method using the `hw6/testAVL.py` script. (You might add additional tests)

Step 2: Run the `hw6/timeBSTandAVL.py` program which adds 5,000 items into initial empty BST and an AVL tree in:
- sorted order, and
- in random order

Record the results in the following table:

	5,000 items added in sorted order		5,000 items added in random order	
	BST	AVL Tree	BST	AVL Tree
Total time to perform adds				
Height of resulting tree				

Step 3: Answer the following questions:
a) Why is the BST so much slower than the AVL tree if we add the items in sorted order?

b) Examine the beginning of the `timeBSTandAVL.py` program. Why did we need to set the recursion limit?

c) Why is the BST faster than the AVL tree if we add the items is random order?

Step 4: Read sections 9.1 and 9.2 of the text. Implement an AVL-tree based implementation of a dictionary called `avlTreeDict`. The dictionary operations are defined in Table 19.3. NOTE: You don't need to implement the `pop` method since we have not talked about deleting from an AVL-tree. You will need to use an `Entry` class simliar to the list-based dictionary implementation (called `ListDict`) on pages 787-788. However, your `Entry` class should also include methods for `__lt__` and `__gt__` (or a `__cmp__` method).

You can test your `avlTreeDict` class thoroughly.

(Note: We will use your `avlTreeDict` class in Homework #7)

Submit all files (avl.py, binarytree.py, binarytreeAVL.py, etc.) as a single zipped file (called hw6.zip) electronically at
https://www.cs.uni.edu/~schafer/submit/which_course.cgi