Data Structures (CS 1520)   Homework #5   Due: 4/17/19 (Wed.) at 11:59 PM
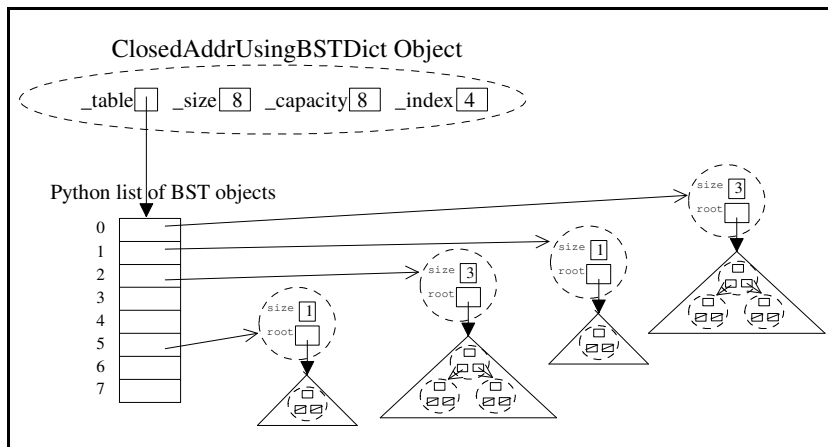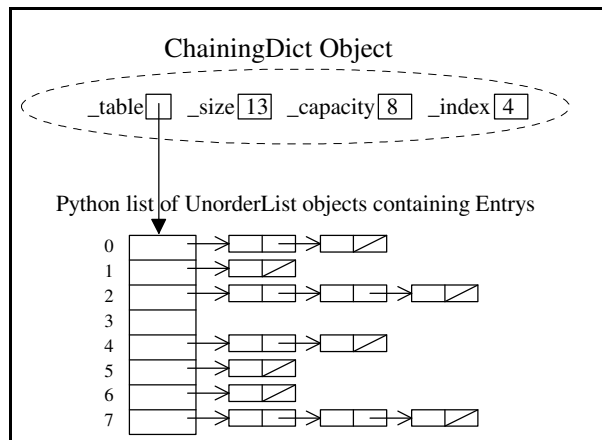
In this assignment I want you to implement a new dictionary by combining binary search trees (BSTs) and closed-address hashing. Recall that the ChainingDict class uses a hash table with UnorderedLists at each slot as shown on the left below. You are to implement a ClosedAddrUsingBSTDict class using a hash table with BSTs at each slot as shown on the right below:

For this assignment you need to implement and test the following subset of dictionary/map methods (Lecture 15 at: http://www.cs.uni.edu/~fienup/cs1520s19/lectures/lec15_questions.pdf).

Method call	Class Name	Description
d =ClosedAddrUsingBSTDict()	__init__(self, capacity = 8)	Constructs an empty dictionary
d["Name"] = "Bob"	__setitem__(self,key,value)	Inserts a key-value entry if key does not exist or replaces the old value with value if key exists.
temp = d["Name"]	__getitem__(self,key)	Given a key return it value or None if key is not in the dictionary
del d["Name"]	__delitem__(self,key)	Removes the entry associated with key
if "Name" in d:	__contains__(self,key)	Return True if key is in the dictionary; return False otherwise
for k in d:	__iter__(self)	Iterates over the keys in the dictionary
len(d)	__len__(self)	Returns the number of items in the dictionary
str(d)	__str__(self)	Returns a string representation of the dictionary

Download hw5.zip from http://www.cs.uni.edu/~fienup/cs1520s19/homework/ . Implement your ClosedAddrUsingBSTDict class in the supplied file closed_addr_using_bst_dictionary.py. Using the timing program in hw5/timeDictionaries.py complete the table on page 2 after finishing your ClosedAddrUsingBSTDict class.

SUBMISSION
Submit **ALL necessary files** to run your ClosedAddrUsingBSTDict dictionary as a single zipped file (called hw5.zip) electronically at

https://www.cs.uni.edu/~schafer/submit/which_course.cgi

Include in your hw5.zip file a "Dictionary Comparison Table" file (.txt, .doc, .rtf, .odt, etc.) containing the completed table on page 2.

Dictionary Comparisons with random integer items (Time in seconds)				
		Number of Integers to Store		
Dictionary Type	**Operations Performed**	**100,000**	**200,000**	**400,000**
Python Dictionary	Insert of Evens			
	Successful Search for Evens			
	Unsuccessful Search for Odds			
Single BST as a Dictionary	Insert of Evens			
	Successful Search for Evens			
	Unsuccessful Search for Odds			
ChainingDict	Insert of Evens			
	Successful Search for Evens			
	Unsuccessful Search for Odds			
Your ClosedAddrUsingBSTDict	Insert of Evens			
	Successful Search for Evens			
	Unsuccessful Search for Odds			