CS/COE 1501 – Algorithm Implementation – Assignment#2[1]

Due: Monday October 19th @ 11:59pm

Late submission: Wednesday October 21th @11:59pm with 10% penalty per late day

OVERVIEW

Purpose: To implement a hybrid Symbol Table that provides hash table access, indexed integer access, and sequential access (using an iterator).

Goal 1: To implement a linear probing hash table.

Goal 2: To implement an iterator.

BACKGROUND

The PHP language has an interesting array data type. Rather than a simple array like that of C or Java, the PHP array is a hybrid of a hash table and a linked list. This allows for a lot of functionality, including hash table access, indexed integer access, and sequential access (via the linked list). For more information on PHP arrays, see: http://php.net/manual/en/language.types.array.php.

DETAILS

In this assignment you will approximate a PHP array using Java. Your basic approach should be as follows:

- You will create a new generic Java class with the following header:
  public class PHPArray<V> implements Iterable<V>

  This will allow a user to instantiate a PHPArray of an arbitrary Java type and to iterate through its values using the predefined Java for loop.

  – Your PHPArray class will be a symbol table and will thus store data as (key, value) pairs, where the keys will be Java Strings and the values will be the parameter type V.

  – Your PHPArray will use a linear probing hash table as part of its underlying storage mechanism. This will enable the put(), get() and unset() [i.e. delete] methods to be done in O(1) time. Your hash table should handle delete and should automatically resize by doubling the

[1] Assignment adapted from Dr. John Ramirez's CS 1501 class.

table size prior to a put() if the table is >= ½ full. For help on this part of your project, see the author's LinearProbingHashST.java class. Note, however, that there will be some differences in your class from this due to the requirement to also keep the data in insertion order via a linked list. For example, I do not recommend keeping the keys and values in separate parallel arrays, like the author does in LinearProbingHashST. Consider rather a linked type (i.e. a node) that contains both key and value fields. These objects could then be stored in your hash table and also accessed sequentially via a linked list.

– Your PHPArray will also use a linked list of data to keep track of the items in sequential order (based on the order that they were put() into the PHPArray). This ordering will be that produced and accessed by the Iterator() and some other methods as well. It is very important that this linked access does not alter the constant time access of the hash table. Thus, you should think carefully about how you will implement this. For example, keeping a completely separate hash table and linked list will not work, since to remove an item from the linked list requires linear time.

– There is a lot of other functionality required of your PHPArray class, based loosely on the functionality of the actual PHP array. This functionality is demonstrated in the main program Assig2.java. See the extensive comments in this file to determine the methods that you must implement and see the output produced in a2out.txt to see the required output. Your PHPArray class must run with this main program and produce the required output.

Important Notes:

- If you are making an array of a parameterized type in Java, there are some issues with the array instantiation that may cause you grief. Below is a link that may help. You can also Google this to see other potential solutions. http://stackoverflow.com/questions/19478225/array-of-generic-nodes-java
- You must utilize the Java Iterable and Iterator interfaces in this project. For help with these see the Java API and also your CS 445 Text.

EXTRA CREDIT

There are a LOT of additional methods / functions available for PHP Arrays. If you implement one or more of these that are non-trivial you can get some extra credit. See:

http://php.net/manual/en/ref.array.php

SUBMISSION REQUIREMENTS

You must submit to **Gradescope** at least the following files:

1) Your PHPArray.java file

2) Any other files that you have written

3) Assignment Information Sheet.

The idea from your submission is that your TA can unzip your .zip file, then compile and run your programs **from the command-line** WITHOUT ANY additional files or changes, so be sure to test it thoroughly before submitting it. If the TA cannot compile or run your submitted code it will be graded as if the program does not work.

If you cannot get the programs working as given, clearly indicate any changes you made and clearly indicate why on your Assignment Information Sheet. You will lose some credit for not getting it to work properly, but getting the main programs to work with modifications is better than not getting them to work at all. A template for the Assignment Information Sheet can be found in the assignment's CourseWeb folder. You do not have to use this template but your sheet should contain the same information.

Note: If you use an IDE such as NetBeans, Eclipse, or IntelliJ, to develop your programs, make sure they will compile and run on the command line before submitting – this may require some modifications to your program (such as removing some package information).

RUBRICS

PHPArray Functionality:	
put()	10
get()	5
unset()	10
put(), get(), unset() with int arguments:	10
Iterator / Iterable:	10
each() method:	10
keys() and values() methods:	10
showTable() method1 : (If this method is not implemented, you may also lose credit on items that cannot be verified without it (ex: unset()))	5
sort() method:	10
asort() method:	10
Sorts throw exception if not Comparable:	5
array_flip() method:	10
PHPArray Implementation Requirements:	
Generic / arbitrary data:	10
Linear Probing Hash Table utilized correctly:	10
Linked List utilized correctly:	10
put(), get(), unset() all O(1):	10
Table resizing as required:	10
Pair<v> class:	10
Documentation:	5
Submission:	5
Extra Credit	10