Activity 8

Actvity 8

In the TableTest repo is a project for testing and graphing the performance of different Table implementions. Your groups task is to implement the Table interface using a simple list approach and also using a binary search tree.

Part 1

Clone the project and set it up so you can compile and run it within your groups development enviroment (don't use Android studio). There is 2 classes with mains in the project the first is "DemoTables" this sets up and runs a few small tests on each of the different table implementations. The second, called "TableTest", which randomally generates large tables of different sizes and times the average performance of the lookup method. Once you are convinced your solution is correct, using "DemoTables, give "TableTest" a go and see how your code performs.

Currently there is only "HashMapTable" Table that is completed, it just uses the standard HashMap class. When you run this code you will see "HashMapTable" working but the others do not work as they are not implemented yet. This lab involves completing their implementations.

However, to get started just read over the code and obtain a basic understanding of how it works and also run it on your computer.

Part 2

Implement the "ListTable" class. This will involve:

- adding fields for storing the table as a linear list (ArrayList is a good class to use, however, you may also just wish to just use an array),
- adding code for the constructor, this will make an empty table,
- adding the code for the insert method (remember if the key is already in the list you need to overwrite it's old value with the new value), and
- adding the code for the lookup method.

To help you out a "KeyValue" class is implemented. This class holds both a key and a value in a single object.

Part 3

Implement the "BinarySearchTreeTable" class. This will involve adding code in a similar way to Part 2. However, you will also need to add another class (or classes) for storing the tree structure. This could be done as extra classes in the project or as inner classes.

Once you have your implementions working and tested with the DemoTables class, explore the performance of your implementation with the "TableTest" class. How do the implemenations compare? What is the shape of the graph as you compare HashMapTable, ListTable, and BinarySearchTreeTable?

Part 4 (optional - challenge)

Have a go at implementing your own version of a hash table. How does your implementation compare to that of the standard one?