CS310: Advanced Data Structures and Algorithmns

Fall 2014 Assignment 3

Due: Thursday, October 2 2014, in class

Goals

practice maps and hash, both theoretically and practically.

Questions

1. What methods of Map can be implemented in O(1) time with a good hash function and a properly-sized hash table? What methods of Set? Could we implement List with a hash table? Explain.

2. Suppose you want to use a HashSet for element objects of a type implemented by some (non-JDK) code youre not supposed to change, yet doesnt have a hashCode() implementation. Suppose it has equals (and you can read the source code for equals) and getters for all fields. What can you do?

3. Weiss, Problems 15.3, 15.4. We need inner and static inner classes (AKA nested classes) for clean HashSet/HashMap implementations and later container classes. Read Weiss, Sec. 15.1 and 15.2, but note that the Iterators there are not generic. It is easy to add type parameters to them, luckily.

4. Practice with nested classes. Fix the SimpleStudent class (pg. 269) as suggested in class 4. Enhance it to have methods addBook(String title, long ISBN) and boolean hasBook(long ISBN) by designing a private static inner class Book (full name SimpleStudent.Book) to hold a book owned by the student, and adding a Collections class field to SimpleStudent to hold all the books owned by the student. In this case Book never gets used outside SimpleStudent. Consider if you need equals(), etc. for Book. Note that at 10 digits, an ISBN (ISBN-10) does not fit in an int.

5. In contrast to 4. modify SimpleStudent to SimpleStudent1 with SimpleStudent1.Book objects that are used outside SimpleStudent1. SimpleStudent1 should have methods addBook(title, ISBN) and SimpleStudent1.Book findBook(ISBN). Make sure Book is sealed up from access by a client of SimpleStudent, and has getters for title, ISBN, and owner name (the students name.) Use a non-static **private** inner class here.

6. Suppose the keys $X_{24}, W_{23}, J_{10}, B_2, N_{14}, S_{19}$ are inserted in that order in an empty hash table T, with the following assumptions:

   - Open addressing (closed hashing),
   - Table T contains space for seven entries, numbered from 0 to 6
   - Hash function of key L_n is $h(L_n)$ = n % 7.

   (a) What is the final configuration of the table if collisions are resolved using linear probing?

   (b) What is the final configuration of the table if collisions are resolved using quadratic probing?

   (c) Why might one of these probing methods be better than the other one?

7. Given below are descriptions of some computer programs. Each of them reads a text file from standard input. For each of them, specify:

- Which APIs would you use. Choose from APIs discussed in class: List/Map/Set
- Which impementations of the APIs would you use, e.g. Linked List / HashTable / Tree ...
- Describe how you would use the API to implement the program. Give pseudocode or describe step by step how your program would work. No need to write an actual program.
- Give the time complexity of your program in terms of the number of words or lines read (whichever is appropriate).

Try to make your programs as simple and efficient as possible. What the programs should do:

(a) Print lines of the file in reverse order.

(b) Print all different words in the file, each word printed exactly once (order not important).

(c) Print all different words in the file, with each word print how many times it occurs in the file (order not important).