

COMP 182 Project 3: Ordering Vectors

(15 points)

Due: 20:00 Monday March 26, 2007

Overview: You will be writing an application to keep track of books in a library. Your program will use several variants of vectors (dynamically allocated arrays). One of the vectors will be an ordered vector (just like the previous project), but the others will insert at the end. For these other vectors, just prior to calling the find or toString methods you will call a sorting method.

For this project you are allowed, but discouraged from using java.util.Vector for the ordered vector portion.

Requirements: You will write 6 classes, an interface, and a file called Analysis.txt.

The first class will be called MyBook. It will contain two String author and title. Instances of this class will be the basic data item (record) which will be stored in the data structures. This class will have at least a constructor, a toString() method, and an accessor getAuthor().

Your next 4 classes will all implement the following interface. You can just copy and paste this into a file called BookVector.java.

```
public interface BookVector {
    public String findAuthor(String str);
    public String findTitle(String str);
    public boolean insert(MyBook bk);
    public boolean delete(String str);
    public String toString();
}
```

The findAuthor returns a String describing all books by a given author. The findTitle returns a String describing all books with a given title. The insert method adds a book to the data structure. The delete method removes the first occurrence of a book by the author passed. The toString method returns a String describing all books in the data structure.

The second class will be called MyBookOrdVec. It will keep the vector ordered at all times and use bisection methods whenever possible.

The third class will be called MyBubbleVec. It will insert new books at the end, delete by “stealing” from the end. Both find methods will be linear searches. However, the toString method will call another method bubbleSort which will sort the vector using bubble sort prior to creating the String.

The fourth class will be called MyMergeVec. It will be just like the previous class except that it calls mergeSort in the toString method.

The fifth class will be called MyRadixVec. It will be just like the previous two classes except that it calls radixSort in the toString method.

The last class will be called MyDriver3. It will instantiate one instance of each type of vector. It will then read from a file called books.txt to initialize the data structures, and then allow interaction between the user and the program. See the sample data file and sample interaction below for additional details. With the exception of using **your** name rather than **Noga**.

Once you have finished programming and debugging your code, time (using `System.currentTimeMillis()`) the 4 structures on a variety of commands. For example, how do the performances compare on: inserting 1000 items in a random order, inserting 1000 ordered items, inserting 500 items and deleting them, inserting 1000 items and then performing 10000 finds. Then write a file `Analysis.txt` about a page long where you describe what you expected your tests to find (and why), what you did find, and why they differed (or not).

Suggestions: If you plan out your classes in detail before coding you will drastically reduce both the length of the code and the amount of debugging. Get the program working with just one of the data structures first. Start as soon as possible and finish programming at least a couple days early (to allow intelligent testing for the `Analysis.txt` file). If you'd like to replace the interface with an abstract class `BookVector`

Submission: When you finish (prior to the deadline), log in to WebCT, go into the Assign Submits area, and upload/submit only the files 6 java files, 1 interface file, and the 1 ascii file. It would be a very good idea to submit the project at least several hours before the deadline (you can resubmit later if necessary). I suggest that when you submit, you request a confirmation email from the server. If you later want to claim the server lost your submission, I will want to see this email.

Grading: This assignment will be graded based upon coding style (comments, indenting, structure, useful var names, etc), whether it compiles and runs properly, how closely your output matches the sample output, whether you followed directions (don't submit class files, don't change the names of the classes, etc), and the quality of your tests and explanation of results.

Sample Data file: The data file will consist of 2 lines for each book.

```
Juan deLucia
Introduction to Weasels
Margarette P. Auspices
Advanced Buffalo Riding
Pat Michael Arronson III
Intermediate Gerbil Farming
Lance N. W. Wingnut
Levitation for Beginners
Lance N. W. Wingnut
Advanced Buffalo Levitation
```

Sample Interaction:

```
jnoga@nogahome:$ java MyDriver3
```

```
Reading data from books.txt
Welcome to Noga Library
Enter a command
> uuummmm?
```

```
Not a valid command!!! Try help.
```

```
Enter a command
> help
```

Legal commands are:

findAuth (displays all books by an author - prefixes allowed)
findTitle (displays all books with a title - prefixes allowed)
add (inserts a book to the data structure)
delete (removes a book from the data structure - prefixes allowed)
show (prints out all books in the library)
help (gives this screen)
quit (ends the program)

Enter a command

> show

NAME	Title
Juan deLucia	Introduction to Weasels
Lance N. W. Wingnut	Levitation for Beginners
Lance N. W. Wingnut	Advanced Buffalo Levitation
Margarette P. Auspices	Advanced Buffalo Riding
Pat Michael Arronson III	Intermediate Gerbil Farming

Enter a command

> findAuth

What is the author's name?

> Lance

NAME	Title
Lance N. W. Wingnut	Levitation for Beginners
Lance N. W. Wingnut	Advanced Buffalo Levitation

Enter a command

> findTitle

What is the title?

> Advanced

NAME	Title
Lance N. W. Wingnut	Advanced Buffalo Levitation
Margarette P. Auspices	Advanced Buffalo Riding

Enter a command

> add

What is the name of the author?

> Jenny Baker

What is the title?

> Sleeping Under a Bright Sky

Enter a command

> delete

What is the name of the author?

> Pat

Enter a command

> show

NAME	Title
Jenny Baker	Sleeping Under a Bright Sky
Juan deLucia	Introduction to Weasels
Lance N. W. Wingnut	Levitation for Beginners
Lance N. W. Wingnut	Advanced Buffalo Levitation
Margarette P. Auspices	Advanced Buffalo Riding

Enter a command

> quit