ECE242 PROJECT 1: Searching in Book Database
Due: September 25, 2014, 11PM on Moodle

Introduction

In the old days, when you wanted to learn about something, you probably needed to ask a teacher or go to a library to find the right books and search all the relevant chapters. With the help of computer technology and Internet, a mouse click will solve all these problems. The Internet is like a mighty teacher who knows everything. The tremendous amount of information and efficient algorithms developed by researchers make this possible. In this project you will have an opportunity to learn the programming skills involved in organizing a book information database from the Book-Crossing community and providing useful search and analysis service.

In the assignment, you will read and manage the book information from an Excel file. You will perform search operations on your organized data structures. By comparing search response times given by different searching methods, you will see the power of good data structures and algorithms.

Task Overview

In this assignment you need to accomplish the following specific tasks.


3. Implement a search method to search for books based on book titles in the book database. In this case, use a binary search algorithm. In order to perform the binary search, sort the books in an alphabetical order based on book title using SelectionSort. Measure the time spent for sorting using System.nanoTime().

4. Generate search queries using ISBN numbers and book titles of 1000 randomly selected books from the database. Illustrate the efficiency of binary search algorithm by comparing
the average time it takes to search based on ISBN numbers and book titles. Measure the
time spent for searching using System.nanoTime().

Getting started

The best way to complete any programming assignment is to take tasks step-by-step. The first
decision you need to make is the number of java classes you need. For this assignment, you
should have at least three classes, including the BookArray class which is provided. Please do
not place all the methods for your entire project in one class. At a minimum, consider creating a
Book class which will call objects of other classes (e.g. BookArray).

Required output

Please provide the following output from your program:

1. Print out book details of the first 10 books in your book database before sorting, including
   the ISBN number, book title, author, year of publication and publisher.
4. Print out book details of the first 10 search queries based on book title (This should be
   the same as the first 10 for the search based on ISBN number).
5. Print out the time required for the SelectionSort.
6. Print out the average times it takes to search based on ISBN numbers as well as on book
titles.

Hints and suggestions

Successfully completing the project and achieving a good grade requires completing the project
as described above and clearly commenting the code. As always, it makes sense to start the
project early. Unless you are an amazing programmer, you probably won’t be able to finish in one
day. Build your project code step by step. For example, verify that you have successfully read in
the database before attempting a linear search. Then, make sure the linear search works before
writing and testing code for SelectionSort, etc.

Each method should have a comment which contains the goal of the method in the beginning
(you can refer to the example of loadBooksToArray method). For key steps in the method, you
may also write some comments to describe their functions. You should name each method or variable properly so that it could be more accessible. Some additional hints are as follows:

1. Make sure that there are no `print` or `println` statements in the code you are attempting to time. The use of these statements will negatively affect recorded time values and lead to incorrect results. Your submitted SelectionSort, linear search and binary search methods should not include these statements.

2. To randomly select 1000 books from the book database, use Math.random() method to generate 1000 random numbers and use these numbers as the indices to retrieve the book titles and ISBN numbers of the 1000 books. It is OK to select the same books multiple times.

3. Do not modify the code portions of the `loadBooksToArray` method that are provided. You can fill in missing code which is noted in the method.

4. The SelectionSort, linear search and binary search methods can follow similar formats as the code used in lectures and discussions.

**What to submit**

Please submit all .java files for your project and the screenshots for all the print out results. All code should be well commented.

*Reminder:* The course honesty policy requires you to write all code yourself, except for the code that we give to you. Your submitted code will be compared with the submitted code from all other students in the course to identify similarities. Note that our checking program is not confused by changed variable or method names.

**Grading**

- Code works (50%)
- Comments (20%)
- Program structure (20%)
- Readability (10%)