PRIFYSGOL ABERYSTWYTH UNIVERSITY

Department of Computer Science

CS21120: Data Structures and Algorithm Analysis
Assignment 1 - **Concordance**

1 Background

Your task for this assignment is to produce a program which can produce a concordance of a document. A concordance is defined by Websters Dictionary from the web site `http://dictionary.reference.com/browse/concordance` as below. I only list the two relevant definitions in this case.

> Concordance Con*cord"ance n. [F., fr. LL. concordantia.]
>
> 3. An alphabetical verbal index showing the places in the text of a book where each principal word may be found, with its immediate context in each place.
>
> His knowledge of the Bible was such, that he might have been called a living concordance. –Macaulay.
>
> 4. A topical index or orderly analysis of the contents of a book.

Your tasks in this assignment are as follows:

1. Read an index file into a data structure;
2. Search a source text for words from the index file;
3. Present an alphabetical list of index words, with the line numbers of where they are found in the source text.

2 The tasks in detail

2.1 Specify and design a data structure used to store the data.

You should use a Hashtable as the primary structure to store the index words, I will not be teaching you about Hashtables until after the assignment is handed in, so you should read about them.

The Hashtable will also have to reference another data structure which will store the line numbers where the words appear.

The choices that you make here will affect the way in which you can process the input data, and you should spend time drawing up a number of options and evaluating them.

2.2 Describe the algorithm(s) you will use to find index words.

Your algorithm to detect the index words must be described in pseudo code or some other form. You will have to make a number of decisions on how to read in the text.

- Will you read in a line at a time, and process each line?
- Will you store a whole file in memory? If you do, what happens with a big input file?
- How will you deal with punctuation?

All these and other decisions will affect your choice of algorithm and the way in which you approach the problem.

For more of a challenge: A concordance may also show context - you might want to store a number of words each side of the index file and re-present them in the output.

2.3 Produce well a structured and commented Java program, with evidence of testing.

You may use some data files from Project Gutenberg at `http://www.gutenberg.org/`. I suggest you start by using something like Oliver Twist, which can be downloaded from `http://www.gutenberg.org/files/730/730.txt`. You will have to compile your own list of index words - you might find it useful to start with a small number of words in the index file, and then test with larger numbers of words when the program is working.

Your program does not have to have a Graphical User Interface, but for maximum marks you should provide one. You could just provide a simple command line interface.

Please ensure that you provide sufficient evidence of testing - even if your program does not work fully, you should still be able to show how and why it doesn't work.

3 The Submission

- You must submit a design for your data structures using UML diagrams where necessary. This should also include the justification for design decisions that you have made;
- You must submit a neatly hand written or typed description of your algorithm, this could be expressed as a flowchart, pseudo-code or another well recognised documentation method;
- You must submit a printed copy of all your Java source code for all new and modified classes, which should be well structured and commented.
- You must submit evidence of testing showing at least one data file being indexed.
- You must also submit an electronic copy of your assignment including all documents as PDFs and code (source and class files) as a zip file to blackboard.

4 The Marking Scheme

This assignment is worth 25% of the marks for the course CS21120, therefore you are expected to spend somewhere around 30 hours working on it. Note also that the majority of the marks will be for getting the concordance generator working - if you only have a nice GUI you will not get very many marks.

Algorithm design 25%

Data structure design 25%

Java code and testing 50%

5 Hand-in dates and times

This assignment should be posted into the Assignment post box located in reception between 10am and 12noon on Monday 31st October 2011. You must include a coursework front sheet signed to declare that you understand and declare originality of your work. This coursework is not anonymous, as you have to hand in code, which is expected to be documented properly with author tags. The electronic submission deadline is at the same time.

Richard Shipman October 18, 2011