BIL 212:Summer 2019

Assignment 1

Assigned Thursday, May 16. You should submit your program by email to ahmetselimkaya@etu.edu.tr by Wednesday, May 29 (any time up to midnight). Remember that your code should be fully documented. I once again remind you to read the academic honesty policy stated in the syllabus.

The goal of this assignment is to use a linked list to count the words from a given text file. You will be using a CIRCULAR doubly linked list. In this type of linked list, there is NO header or trailer sentinels (dummy nodes). The address of the first node is in a variable called **head**. The **previous** instance variable of the head node contains the address of the last node. The **next** instance variable of the last node contains the address of the first node.

Each word and its frequency (number of times it appears in the given text) will be stored in one node of the linked list. The nodes must be kept sorted in decreasing order of frequencies. If two words have the same frequency, they are sorted alphabetically.

When a new word is read from the file, you need to add the word into the list. If the word already exists in the list, you simply will increment the value of "frequency" for that word. Since the words are kept in sorted order in the linked list by their frequency order, when the frequency of a word has changed, you have to move the word to correct position in the linked list.

You are NOT allowed to use Java's LinkedList class. The DoublyLinkedList.java posted on our class page on Piazza is NOT circular, hence you can NOT use that class directly either.

When you are parsing input files ignore punctuation and capitalization.

Input Files: Your program will receive input from two different files.

- **text-file:** The text file from which the words are read and counted.
- **directives-file:** This file will contain one directive per line. Each directive consists of the name of the directive followed by 0 or more arguments, separated by spaces. You may assume that only the directives given below will appear in the file, and the specified arguments will always appear. The output of the directives will be written to standard output.

Directives: Here are the possible directives and their meanings.

- `load filename`
  Parse the words in the file named `filename` and construct the frequency linked list.
- `print-max N`
  Print the `N` highest frequency words and their frequencies in the list, one word per line and sorted highest to lowest. If N is larger than the number of words in the list, print the entire list.
- `print-min N`
  Print the `N` lowest frequency words and their frequencies in the list, one word per line and sorted lowest to highest. If N is larger than the number of words in the list, print the entire list.

- `print-range N1 N2`
  Print the words whose frequency is within the range `[N1, N2]`. Print each word and its frequency, one word per line and sorted highest to lowest. If no words are in that range, print 'This range is empty'.

- `print-freq word`
  Print the frequency of the given `word`.

- `print-nth N`
  Print the word whose frequency is the `N`-th highest in the list. For example if N=1 you print the word with the highest frequency, if N=2 you print the second highest, etc.

- `truncate-list N`
  Remove the N lowest frequency words from the list. If N is larger than the number of words in the list, clear the entire list.

Important: The program will be invoked from the command line as: `java word-frequency <directives-file>`