

ENSF 594 – Principles of Software Development II

Summer 2021

Lab Assignment # 3

Linked Lists

Total Marks: 25

Due Date: Monday, July 26, by 11:59 PM using D2L

Linked List

The goal of this assignment is to write a Java program that arranges a list of words into separate lists of anagrams. The input is a text file that contains a list of words, each word on its own line. The number of words in the input is arbitrary and could be very large. The program should print to an output file the lists of anagrams in the following way:

- All the words that are anagrams of each other are displayed together on a single line; any word without any corresponding anagrams is displayed alone on a line.
- Words in a line should be separated by a single space or comma.

For example, this input:

car

dog

bed

stop

god

pots

arc

tops

yield the output:

arc car

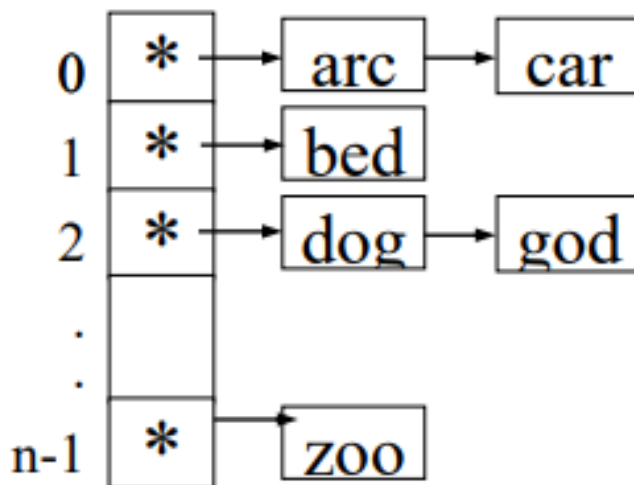
bed

dog god

pots stop tops

The order can be different.

You must use linked lists to deal with the arbitrary number of anagrams in a line and use an array of references to keep track of all the linked lists. For example:



An acceptable alternative to using an array is to use a vector or ArrayList. You can use a class such as Vector or ArrayList from the Java libraries for this).

You must write your own implementation of linked lists.

One way to determine if two words are anagrams is to sort the letters in both words. If the sorted words are the same, then the original two words are anagrams of each other. For example, “dog” and “god” are both sorted into “dgo”, so they are anagrams.

Write a program in Java to implement the above requirements. You can assume that your input file name is “input.txt” and output file name is “output.txt”.

Include a readme file to write how to run your code.

Complexity Analysis:

1. What is the worst-case complexity of your algorithm when checking if two words are anagrams of each other? Express this using big-O notation, and use the variable k to represent the number of letters in each word.

Note: It is suggested to write implementation of the code. If you are using a pre-defined sort method, you are not able to identify the complexity.

Submit electronically via D2L:

1. A pdf report answering the question
2. The readme file.
3. Your source code files. Your TA will run your program to verify that it works correctly. Make sure your Java program compiles and runs without issues.

Grading Rubric:

Functionality: produces correct output: <ul style="list-style-type: none">• No anagrams words in the text files (5 marks)• Anagrams words in the text files (10 marks)
Readability: code is clear and easy to read (4 marks)
Documented: code classes/functions/inline commented (2 marks)
Report: Complexity analysis (4 marks)