Seneca College
Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444 **Submission date:** **April 05, 2023**

Workshop 9

Description:
The following workshop lets you practice basic java coding techniques, creating classes, methods, using arrays, inheritance, polymorphism, Exceptional Handling, Java I/O, JavaFx, lambda's, collection framework.

Task - 1 (Medium to Hard Challenge)

Write a program that computes the edit distance (also called the Levenshtein distance, for its creator Vladimir Levenshtein) between two words. The edit distance between two strings is the minimum number of operations that are needed to transform one string into the other. For this program, an operation is a substitution of a single character, such as from "brisk" to "brick". The edit distance between the words "dog" and "cat" is 3, following the chain of "dot", "cot", and "cat" to transform "dog" into "cat". When you compute the edit distance between two words, each intermediate word must be an actual valid word.

Edit distances are useful in applications that need to determine how similar two strings are, such as spelling checkers.

Read your input from a dictionary text file. From this file, compute a **map** from every word to its immediate neighbors, that is, the words that have an edit distance of 1 from it. Once this map is built, you can walk it to find paths from one word to another.
A good way to process paths to walk the neighbor map is to use a **linked list** of words to visit, starting with the beginning word, such as "dog".
Your algorithm should repeatedly remove the front word of the list and add all of its neighbors to the end of the list, until the ending word (such as "cat") is found or until the list becomes empty, which indicates that no path exists between the two words.

Note: you can read more about the Levenshtein Distance, the following article also discuss different algorithms in order to calculate the distance between words.

Continue to the next page…

Example run:
Words in the file are

dog
dot
cot
cat
fat
mat
rat
rut

```
This program uses a dictionary to compute the
edit distances between pairs of words.

What is the dictionary file? test.txt

Let's find an edit distance between words.
    first word (enter to quit)? dog
    second word? cat
dog, dot, cot, cat
Edit distance = 3

Let's find an edit distance between words.
    first word (enter to quit)? dog
    second word? rut
dog, dot, cot, cat, rat, rut
Edit distance = 5

Let's find an edit distance between words.
    first word (enter to quit)? q
q is not in the dictionary

Let's find an edit distance between words.
    first word (enter to quit)?
```

Workshop Header

/***
Workshop #
Course:*<subject type> - Semester*
Last Name:*<student last name>*
First Name:*<student first name>*
ID:*<student ID>*
Section:*<section name>*
This assignment represents my own work in accordance with Seneca Academic Policy.
Signature
Date:*<submission date>*
***/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly using JavaDoc
- JavaDoc should be generated properly in the project
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- You are supposed to create **video with voice** of your running solution. **(50%)**
  - Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is)

    OR
  - Show your work during the lab
- A text file which will reflect on learning of your concepts in this workshop. **(20%)**
  - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt
- JavaDocs must be used for proper documentation of each task. **(15%)**
- Submission of working code. **(15%)**
  - Make sure your follow the "**Code Submission Criteria"** mentioned above.

    - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.**zip.
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.

Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.