**EMAN FATIMA**

**BSCS 4C**

**34648**

ADV. Programming

ASSIGNMENT 1

WORD LADDER PROBLEM

**Introduction to Word Ladder:**

A Word Ladder is the connection between 2 same letter words, such that each transition to a step can only be made by replacing only 1 letter in the source word and the target word is always a meaningful word. A good solution has a number of steps equal to the number of places the given words differ in (Hamming Distance).

**Solution:**

The solution involves saving the dictionary in the form of a hash map such that each key is made of a word with one letter as wildcard that can be replaced by multiple letters to form a meaningful word. All such words are mapped against each key. The hash map is used to create a graph which in turn is used to keep track of all the connections between words(nodes). Breadth-First-Search is further applied to this graph and provided a start Word(starting node) and end Word(end condition/node). The path obtained is the optimal path between two given words.

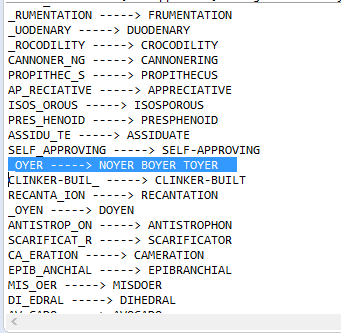
**The Coding Process:**

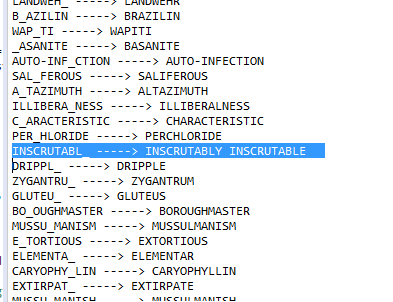
**Step1:** Creating required classes to store Dictionary data and to provide user interface as well as to perform operations on class objects.

**Step 2:** Reading words from a JSON file.

**Step 3:** Storing words read in Hash Map<String, ArrayList<DictData>>

Where keys are words with a wildcard letter each and each key has an arrayList (bucketed list) connected to it which has nodes of the type Class DictData()

****

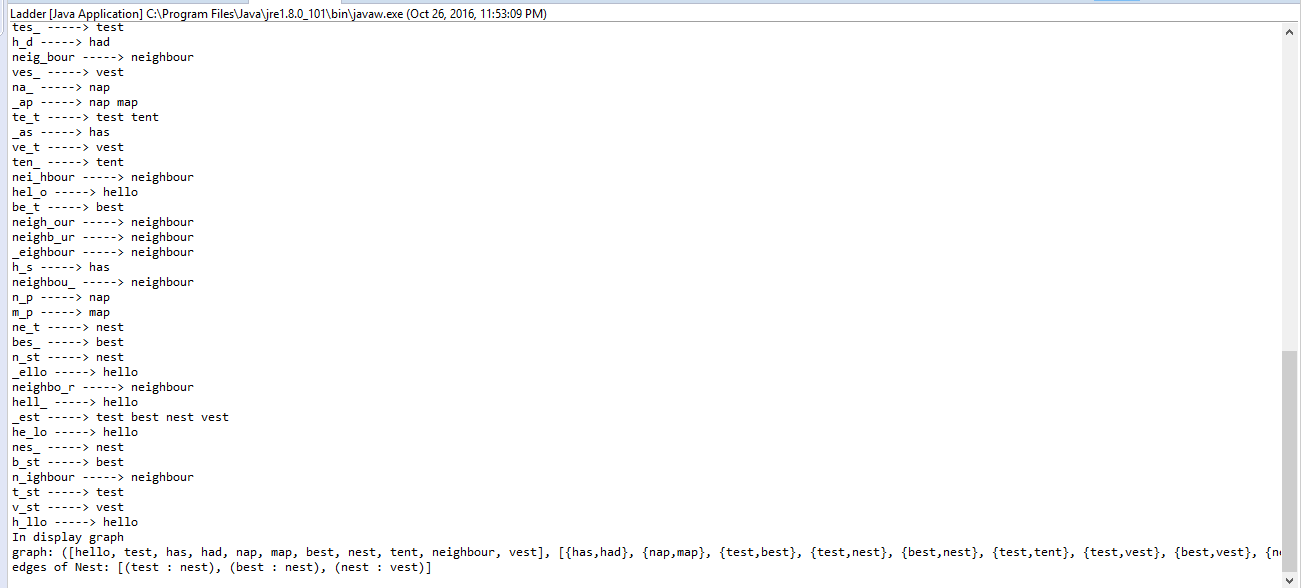
****

**Step4: Creating a Graph**

*Graph Printed as String:*

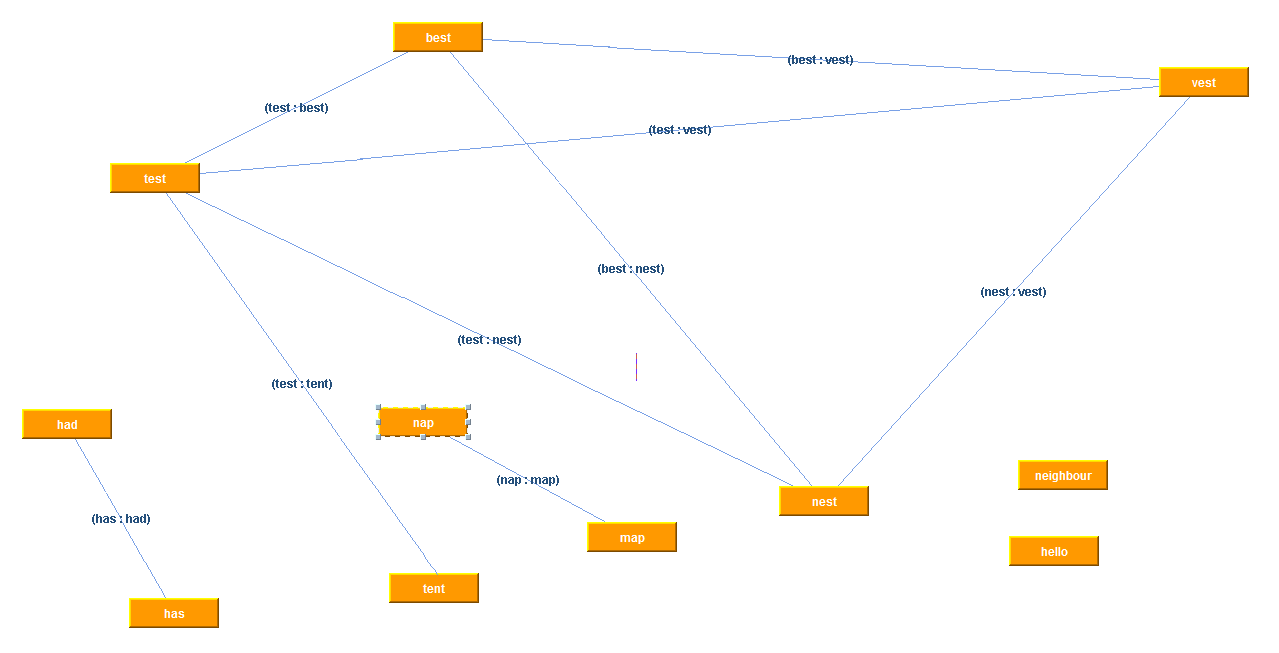
At the bottom of the screenshot, a graph has been displayed as string.

The graph has been created using JGrapht library.

****

**Step 5: Displaying Graph**

*The test graph has been visualized using Jgraph library.*



**Step 5: Applying Breadth-First-Search to Graph:**