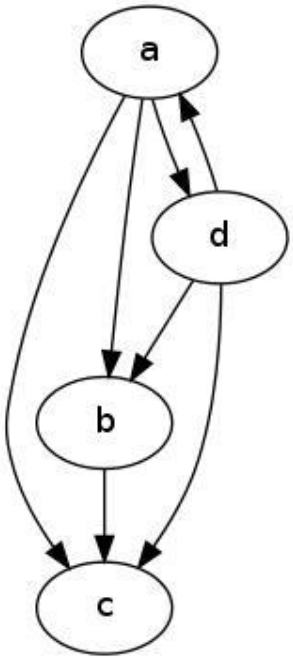
# CSCI 3005 – Programming Assignment 3 – Spring 2018




In English, some words are commonly followed by other words, giving the pair of words a special meaning. Sample word pairs include expressions such as "poetic justice" and "egg white." The fact that a given word w is commonly followed by another word v can be represented as a directed graph containing the edge (w, v).

Your task is to use the breadth-first search algorithm studied in class to perform various operations on a graph of word pairs. In developing your solution, you are to use the `DiGraph.java` class provided by the instructor. Data for your program will be obtained from text files in which each line contains two words representing a valid word pair in the graph.

Your solution is to be implemented as a class named `WordPairs` containing the following public methods:

`WordPairs(String filename)`: a constructor that reads in the data from a text file which contains a series of lines. Each line has two words separated by a single space.

`String wordChain(String first, String last)`: returns the shortest sequence of word pairs that begins with *first* and ends with *last*, using the format below (returns `[]` if there is no such sequence).

```
[first word1,word1 word2, …,wordn last]
```

`int chainLength(String first, String last)`: returns the number of word pairs in the shortest chain that begins with *first* and ends with *last*. Return `Integer.MAX_VALUE` if none exists.

`int reachableFrom(String word)`: returns the number of distinct words that are part of all chains that begin with *word*. Assume that a word is always reachable from itself.

`int reachableFrom(String word, int maxLength)`: returns the number of distinct words that are part of all chains of *maxLength* that begin with *word*. For instance, given the sample graph above, the method call `reachableFrom("a", 1)` would return 4.

`String reachableWords(String word, int maxLength)`: returns a String containing the distinct words that are part all chains of *maxLength* that begin with word. The reachable words should be grouped by level, using the format

```
[word]
[word1, word2]
[word3, word4, word5]
 …
[…,…, wordn]
```

`String cycle(String word)`: returns the shortest sequence of word pairs that begin and ends with *word*, using the format below (returns `[]` if there is no such sequence).

```
[word word1,word1 word2, …,wordn word]
```

The WordPairsTest.java program and sample text files have been provided for partial testing. Submit your WordPairs.java (and any other .java files developed as part of your solution) to Mimir for testing. You should submit neither DiGraph.java nor any data files nor any .class files.