

Anagram Finder README

How to use

To use, run the jar `AnagramFinderMain.jar` located in the root folder. The program can be run in several different ways:

1. **`java -jar AnagramFinderMain.jar filepath [-cs] {words}`**

Prints anagrams for given word(s) to std out. Words should be separated by spaces.

2. **`java -jar AnagramFinderMain.jar filepath [-cs]`**

Starts a REPL into which the user can enter one or several words for which he/she wants to find anagrams. Words should be separated by spaces. Press return on an empty line to exit the REPL.

3. **`java -jar AnagramFinderMain.jar filepath -g [-cs]`**

Starts the UI version of the anagram finder.

In all of the above, the first argument should be the path to the text file to be used as the dictionary. The `-cs` flag specifies whether or not the program should care about case sensitivity.

Implementation Details

The anagram finder is implemented as a hashtable. The values in the table are lists of strings which have the same number of each letter, and the keys are the corresponding alphabetically sorted string representing each list. For example $\langle abc, \{abc, bac, cba, acb\} \rangle$ is a valid key-value pair in the table.

To add a string to the word bank, the program sorts the string, then adds the string to the list in the appropriate bucket. To find the anagrams of a string, the program sorts the string, and returns the list in the appropriate bucket.

If n is the number of words in the dictionary, and m is the length of the longest string in the dictionary, constructing the table requires $O(nm \log m)$ time, and $O(n)$ space. Finding the anagrams of a word of length m requires $O(m \log m)$ time.

Tests

Unit tests are located in the `/test` folder.