

THE ANAGRAMS PROBLEM

Two words are anagrams of each other if they have the same letters in the same quantity, in a different sequence. For instance, **cat** and **act** are anagrams. Also **pots**, **stop**, and **tops** are all anagrams of one another (each can be formed by permuting the letters of the others).

TASK 1: Programming assignment

Write a program that, given a word, will find all anagrams of the word that are in the provided dictionary (*words.txt*). The program should repeatedly ask the user to enter a word and it should display the anagrams of the word entered by the user. The program should also give the user the option to exit.

Sample run:

```
Enter a word (press enter to exit)> cat
Anagrams of cat: act cat
Enter a word (press enter to exit)> meat
Anagrams of meat: mate meat meta tame team
Enter a word (press enter to exit)> pot
Anagrams of pot: opt pot top
Enter a word (press enter to exit)>
Good bye
```

Implement ONE of the following solutions to the problem:

SOLUTION 1

At the beginning of the program, load the dictionary into a **map** where the *key* is a word in its alphabetized form and the *value* is the set of all words that map to that *key*. A word is converted to its alphabetized form by sorting all the letters in the word alphabetically. The *map* will look like:

<u>KEY</u>	<u>VALUE</u>
act	act, cat
aeonrsst	senators, treasons
deirw	weird, wider, wired, wried

When looking for the anagrams of a word, lookup the alphabetized form of that word in the *map* and then simply retrieve the set of words associated to that *key*.

SOLUTION 2

At the beginning of the program, load all the words from the dictionary into a **set**.

When looking for the anagrams of a word, compute all permutations of the word, that is, all possible arrangements of the letters in the word, for example, for *cat*, the permutations are *cat*, *cta*, *act*, *atc*, *tac*, *tca*. As you compute each permutation, check whether it is a valid word by searching for it in the **set**. The permutations found in the **set** are anagrams of the input word.

Task 2: Running Time Analysis

Provide and briefly explain the big-O time of your solution.