Homework #5

Due: Friday, December 7, 2012 at 11:55PM **Points:** 100

Please turn in your answers for the homework assignment on SmartSite, under Homework #5 in Assignments. Turn in your programs answers for the extra credit under Extra Credit #5 there.

1. (*100 points*) *Anagrams* are words formed with the same letters. For example, “last”, “salt”, and “slat” are anagrams, because they all contain exactly the same letters: “a”, “l”, “s”, and “t”. If a letter is repeated, the anagrams will have the same number of repetitions; so “aboard” and “abroad” are anagrams, but “board” and “aboard” are not.

   The file “dict.txt”, available in the Text folder of the Resources area on Smartsite, is a list of words, one per line. You are to ask the user for a word, and then print all the word in that file that are anagrams of the word the user enters. Your program is to terminate when the user types control-d (end of file).

   The input may include blanks and other non-letters, so you should treat the line of input as what to look for (the “word” in what follows). Also, treat upper and lower case the same.

   The rules for output are:

   (a) If the user enters a word not in the word list, the program prints:
   `The word` *`word`* `is not in the dictionary`

   (b) If the user enters a word in the word list for which there are no anagrams, the program prints:
   `There are no anagrams for` *`word`*

   (c) If the user enters a word in the word list for which there is exactly one anagram, the program prints:
   `The only anagram for` *`word`* `is` *`anagram1`*

   (d) If the user enters a word for which there are more than one anagram, the program prints:
   `The anagrams for` *`word`* `are` *`anagam1`* `and` *`anagram2`*
   or, if there are more than 2 anagrams, the program prints:
   `The anagrams for` *`word`* `are` *`anagam1, anagram2`* `and` *`anagram3`*
   (or however many anagrams there are for the word).

   Here is some sample output:

```
Your word? aboard↵
The anagrams for aboard are aborad and abroad
Your word? welcome↵
There are no anagrams for welcome
Your word? phe↵
The word phe is not in the dictionary
Your word? hep↵
There are no anagrams for hep
Your word? salt↵
The anagrams for salt are last and slat
Your word? alst↵
The word alst is not in the dictionary
Your word? gear↵
The anagrams for gear are ager, agre, gare and rage
Your word? control-d↵
```

   Please call your program “anagram.py”.

   Hint: Store the words in a dictionary. Try sorting their characters into order and using that as a key. Then you can store words with the same characters as a list, accessed by the key produced by sorting their characters.

Extra Credit

Remember to hand this in under Extra Credit #5, *not* under Homework #5!

2. (*30 points*) The "rank" of a word is its position in a list of words sorted by frequency. The most common word has rank 1, the second most common rank 2, and so forth.

   Zipf's law describes a relationship between the ranks and frequencies of words in natural languages. It predicts that the frequency f of the word with rank r is

   $$f = cr^{-s}$$

   where s and c are parameters that depend on the language and the text. If you take the logarithm of both sides of this equation, you get

   $$\log f = \log c - s \log r$$

   So if you plot $\log f$ versus $\log c$, you should get a straight line with slope $-s$ and intercept $\log c$.

   Write a program that prompts the user for a file name, reads text from that file, counts word frequencies, and plots $\log f$ against $\log r$ using the `turtle` package. Check whether this plot is a straight line. If it is, can you estimate s?

   Please use the file "alice.txt" for this problem. It is available in the Text folder of the Resources area on SmartSite.

   Please call your program "zipf.py".