

Anagrams

- An anagram of a word is another word which uses the same letters but in a different order. For instance, “maples” and “sample” are anagrams (of each other), but “abb” and “aab” are not, because one has 2 b’s and other 2 a’s (even if they were actual words). Other anagram pairs include “absorption”/“probations”, and “abides”/“biased”. There are, in fact, entire anagram families.
- Here’s an anagram family: *limes miles slime smile*
- You should have the file “dictall.txt” of about 80,000 English words on your computers.
- Create the function *IsAnagram(word1, word2)* which will be given 2 words, and will return True if they are anagrams of each other. Test it on any anagram pairs that you know, including the ones above (also test it on “abb” and “aab”, to make sure it returns False).
- Now create the *AnagramFamily(filename, member)* which will be given the filename of a file containing a list of English words, and also a single word (*member*), and will return the list of all words in the wordlist which are anagrams of the *member* word, namely the anagram family containing the *member*. For example:
AnagramFamily('dictall.txt', 'miles') returns [*'limes', 'miles', 'slime', 'smile'*]
- All right, what is the anagram family containing:
 - “parties”?
 - “mental”?
 - “teardrop”?
 - “triangulating”?
 - “antagonist”?
- The size of the *AnagramFamily* containing “smile” (above) is 4. Find the size of the largest *AnagramFamily*. Are there other families of the same size? What are all the families of anagrams of that maximum size?
- Here’s a poorly specified problem: this entire adventure in anagrams started when someone asked me for an anagram of “CARTHORSE”. That turned out to be an interesting non-obvious word. Find some examples of “interesting”, “unusual” anagrams. There’s no precise definition for “interesting” and “unusual”, so use your Python tools to explore these anagram families, eyeball some results and see what intrigues you (this is how interesting puzzles are created).