

BIS 420 PROGRAMMING FOR DATA SCIENCE
PRAJAKTA POHARE
CHAPTER 12 EXERCISE 12.4
ILLINOIS STATE UNIVERSITY

More anagrams!

1. Write a program that reads a word list from a file (see Section 9.1) and prints all the sets of words that are anagrams. Here is an example of what the output might look like:

```
['deltas', 'desalt', 'lasted', 'salted', 'slated', 'staled']  
['retainers', 'ternaries']  
['generating', 'greatening']  
['resmelts', 'smelters', 'termless']
```

Hint: you might want to build a dictionary that maps from a set of letters to a list of words that can be spelled with those letters. The question is, how can you represent the set of letters in a way that can be used as a key?

2. Modify the previous program so that it prints the largest set of anagrams first, followed by the second largest set, and so on.

3. In Scrabble a “bingo” is when you play all seven tiles in your rack, along with a letter on the board, to form an eight-letter word. What set of 8 letters forms the most possible bingos? Hint: there are seven.

Solution: http://thinkpython.com/code/anagram_sets.py.

```
from __future__ import print_function, division
```

```
def signature(s):
```

```
    t = list(s)  
    t.sort()  
    t = ".join(t)  
    return t
```

```
def all_anagrams(filename):
```

```
    d = {}  
    for line in open(filename):  
        word = line.strip().lower()  
        t = signature(word)  
        if t not in d:  
            d[t] = [word]  
        else:  
            d[t].append(word)
```

```

return d

def print_anagram_sets(d):
    for v in d.values():
        if len(v) > 1:
            print(len(v), v)

def print_anagram_sets_in_order(d):
    t = []
    for v in d.values():
        if len(v) > 1:
            t.append((len(v), v))
    t.sort()
    for x in t:
        print(x)

def filter_length(d, n):
    res = {}
    for word, anagrams in d.items():
        if len(word) == n:
            res[word] = anagrams
    return res

if __name__ == '__main__':
    anagram_map = all_anagrams('/Users/prajaktapohare/Library/CloudStorage/OneDrive-ILStateUniversity/BIS420/Week 12/words.txt')
    print_anagram_sets_in_order(anagram_map)
    eight_letters = filter_length(anagram_map, 8)
    print_anagram_sets_in_order(eight_letters)

```

Users > prajaktapohare > Library > CloudStorage > OneDrive-ILStateUniversity > BIS420 > Week 12 > BIS420_PrajaktaPohare_Ch12_12.4.py > signature

```
1  from __future__ import print_function, division
2
3  def signature(s):
4      t = list(s)
5      t.sort()
6      t = ''.join(t)
7      return t
8
9  def all_anagrams(filename):
10     d = {}
11     for line in open(filename):
12         word = line.strip().lower()
13         t = signature(word)
14         if t not in d:
15             d[t] = [word]
16         else:
17             d[t].append(word)
18     return d
19
20
21 def print_anagram_sets(d):
22     for v in d.values():
23         if len(v) > 1:
24             print(len(v), v)
25
26
27 def print_anagram_sets_in_order(d):
28     t = []
29     for v in d.values():
30         if len(v) > 1:
31             t.append((len(v), v))
32     t.sort()
33     for x in t:
34         print(x)
35
36
37 def filter_length(d, n):
38     res = {}
39     for word, anagrams in d.items():
40         if len(word) == n:
41             res[word] = anagrams
42     return res
43
44
45 if __name__ == '__main__':
46     anagram_map = all_anagrams('/Users/prajaktapohare/Library/CloudStorage/OneDrive-ILStateUniversity/BIS420/Week 12/words.txt')
47     print_anagram_sets_in_order(anagram_map)
48     eight_letters = filter_length(anagram_map, 8)
49     print_anagram_sets_in_order(eight_letters)
50
```