

Lecture 19: Dictionaries

Creating token from a text file:

```
1 def file_to_tokens(filename):  
2     with open(filename) as fin:  
3         return fin.read().split()
```

Create token counts for each unique token:

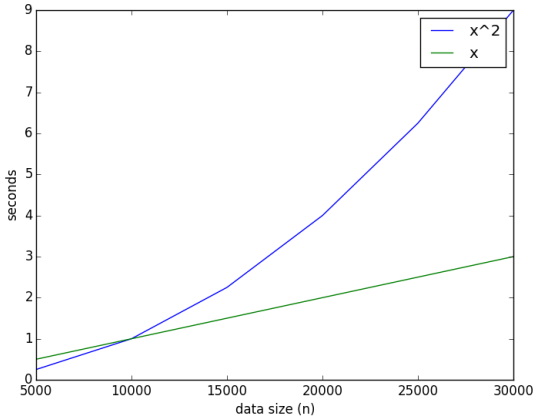
```
1 def wc_list(tokens):  
2     uniq = []  
3     for token in tokens:  
4         if token not in uniq:  
5             uniq.append(token)  
6     return [(t, tokens.count(t)) for t in uniq]
```

```
>>> cProfile.run('wc_list(first5000)')
4575 function calls in 0.238 seconds
```

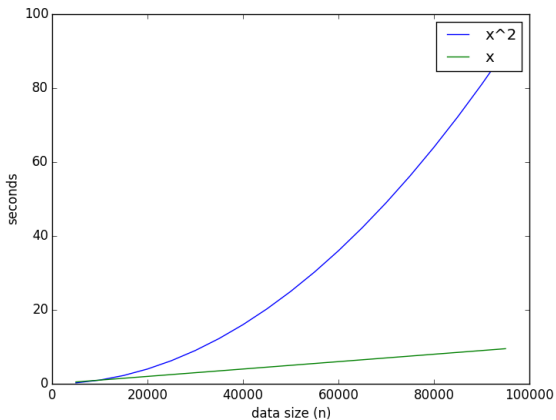
Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.238	0.238	<string>:1(<module>)
1	0.060	0.060	0.238	0.238	freq.py:12(wc_list)
1	0.001	0.001	0.177	0.177	freq.py:18(<listcomp>)
1	0.000	0.000	0.238	0.238	{built-in method builtins.exec}
2285	0.000	0.000	0.000	0.000	{method 'append' of 'list' object}
2285	0.176	0.000	0.176	0.000	{method 'count' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Pro

Quadratic versus Linear



Quadratic versus Linear



```
1 def wc_dict(tokens):
2     counts = {}
3     for token in tokens:
4         if token in counts:
5             counts[token] += 1
6         else:
7             counts[token] = 1
8     return counts.items()
```

Suppose we wanted to create an index of the positions of each token in the original text. Write a function called `token_locations` that, when given a list of tokens, returns a dictionary where each key is a token and each value is list of indices where that token appears.

```
>>> l = "brent sucks big rocks through a big straw".split()
>>> print(token_locations(l))
{'big': [2, 6], 'straw': [7], 'brent': [0], 'a': [5],
 'through': [4], 'sucks': [1], 'rocks': [3]}
```