

# Advanced Homework 11

Assigned: Friday, March 25, 11:00AM

**Due: Before the first lecture on Friday, April 8**

## Submission Instructions

To receive credit for this assignment you will need to stop by someone's office hours, demo your running code, and answer some questions.

### 1 Topic Discovery

Write a script that takes a search term, an integer, and any number of files on the command line, and prints out which files have the search term at least that many times.

For example:

```
$ cat file1.txt
I love algorithms! Analyzing algorithmic complexity is my favorite field of CS.

$ cat file2.txt
This is a file with only one use of the word algorithm.

$ cat file3.txt
I hate algorithms!
I dislike algorithms so much that I have written this treatise against algorithms.

$ python topic.py algorithm 2 file1.txt file2.txt file3.txt
file1.txt
file3.txt
```

## 2 Iterator Protocol

As a very dynamic language, Python has a concept called “protocols”. The idea here is that language constructs, such as a for loop, are really just calling some well-defined functions under the hood.

Iterable objects in Python are things that make sense to loop over in a for loop, for example strings are iterable:

```
>>> for letter in "abc":
...     print(letter)
...
a
b
c
```

As are tuples:

```
>>> for number in (1, 2, 3):
...     print (number)
...
1
2
3
```

Many functions, such as sum also accept iterable objects:

```
>>> sum((1,2,3))
6
```

Any object can be iterated if it implements a `__iter__` function (similar to `begin()` on a C++ STL container) and a `__next__` function (similar to the `++` operator on a C++ STL iterator).

<http://stackoverflow.com/questions/9884132/what-exactly-are-pythons-iterator-iterable-and-ite> gives another summary of Python iterators and pointers to some more resources you may find helpful.

Modify your script from Question 1 such that the following works:

```
>>> import glob # This is a built-in python package
>>> import topic
>>> searcher = topic.Searcher()
>>> for filename in searcher.search('algorithm', 2, glob.glob('*.*txt')):
...     print(filename)
...
file1.txt
file3.txt
```

### 3 Git Pre-Push Hook

Suppose we're writing a program `prog`, and have constructed a suite of regression tests that are named `test*.cpp`, where `*` is the wildstar (any string may be in its place). Write a script that:

1. Compiles the main program, you may compile it how you like. Stop if there are compilation errors.
2. Compiles all the test cases, again compile as you like and stop if there are compilation errors.
3. Run the test case and provide feedback on how many succeeded and failed.
4. If any test case fails, stop the push.