CS 283
Spring 2013

Homework 2: Ruby

Grep or Webgrep

Due beginning of class Thursday, February 28th

Ruby is a higher-level scripting language with many powerful features and easily-installed libraries. Some of these features include code blocks, open classes, easily defined regular expressions, and extensive I/O libraries. In this assignment you will be utilizing these Ruby language features to implement a text searching utility.

You will have two options for this assignment. The first (and easier) option is to write a program that searches a given directory or file for a given string or regular expression, and prints the file names and line numbers where the string is found. This option is to behave like the Unix command line utility "grep", and will be described in more detail under the Grep section below. The second (and more difficult) option is to write a program that searches a given web page and the tree of sites linked from that web page for a given string or regular expression. This option is described below in the Webgrep section.

Since there are two options for this assignment, I expect that you will decide for yourself which one to pursue. But, I do expect more advanced students to take on the more difficult problem. If you have taken at least 3 programming courses previous to this one, this almost definitely applies to you. Talk to me if you are unsure which assignment to pursue.

Whichever assignment you choose, when you are done your program will:

- Have your name at the top of the file.
- Have the version of Ruby that you used at the top of the file.
- Implement the program well.
- Exhibit excellent programming style.
- Have appropriate commented documentation.

You will submit both your program and report electronically through the course's website. As will always be the case, your report should include descriptions of the language features you took advantage of, snippets of code illustrating those features, comparisons with other languages you know, and information about why you chose the program design that you did.

Grep

For this option of the assignment, you will write a program that searches for a given string or regular expression within a directory tree. Your program, grep.rb, will be invoked as follows:

Finds all occurrences of `"File"` or `"file"` in current directory tree:

```
ruby grep.rb "(F|f)ile" ./
```

Finds all occurrences of numbers in the directory tree of the directory "`dir`":

```
ruby grep.rb "\d+" dir/
```

Your program should print the filename, line number, and line of every occurrence of the target regular expression within the given directory tree. The output should be equivalent (at least in information) to the Unix command "`grep -r -n "string" dir/`", except that it should also accept Ruby-formatted regular expressions.

To complete this assignment, I recommend that you take advantage of Ruby's open classes to add grep_rn(target) methods to Ruby's File and Dir classes. This method should recursively look in every file and subdirectory in the given directory and print out the filename, line number, and line of any line that matches the target regular expression.

Webgrep

For this option of the assignment, you will write a program that searches for a given string or regular expression within a given number of links starting at a given web page. Your program, webgrep.rb, will be invoked as follows:

```
ruby grep.rb "(H|h)elmuth" http://people.cs.umass.edu/~thelmuth/index.html 3
```

This command finds all occurrences of "`Helmuth`" or "`helmuth`" in the tree of web pages rooted at the URL `http://people.cs.umass.edu/~thelmuth/index.html` that can be reached by depth 3 (i.e. within clicking 2 links). The program prints each URL where the given string or regular expression was found. Unlike the grep assignment above, you simply need to print the URLs containing the target, not the line containing each instance of it.

In order to complete this assignment, you will need to use a number of built-in and downloaded Ruby libraries. I recommend that you use Ruby's open classes to add methods to the Nokogiri::HTML::Document class. This will require you to download and install the Nokogiri library from the internet using Ruby gems. You will also likely need to use some methods from the open-uri library, which is a Ruby standard library.

There are a number of interesting problems that you should consider while completing this assignment:

- "Trees" of web pages often contain cycles. Your program should only visit each page once, which will greatly increase your program's speed over one that visits pages many times.
- Some web pages contain broken or invalid URLs, may be not publicly available, or may contain unrecognized character formats. Your program should handle these situations gracefully without crashing or giving error messages.
- Text may appear in a variety of places in a web page. Your program should only search for the target string outside of any HTML tags. For example, if a page contains `<span class="bad">good</span>`, then your program should return the page when searching for "good", but not when searching for "bad". The Nokogiri::HTML::Document class

has a variety of methods that will help you with finding links and finding text that is outside of tags.