Project Assignment (Due 12/13):

Project Assignment Due 12/13

Attached Files: Project Assignment and links.db. You will need the attached database file. The project is just a series of related questions from the textbook involving ranking of web pages, different methods used to give a rank, and different implementations: Perkovic Programming Problems, Chapter 12, #16, 19, 24, 25. You will submit the .py files with no syntax errors. You must also submit a (short) document file explaining why one ranking method might be "better" than another, and comparing the different implementations

#####Project Assignment Background Help!!!!!#####

Figure 12.1 Five link web pages

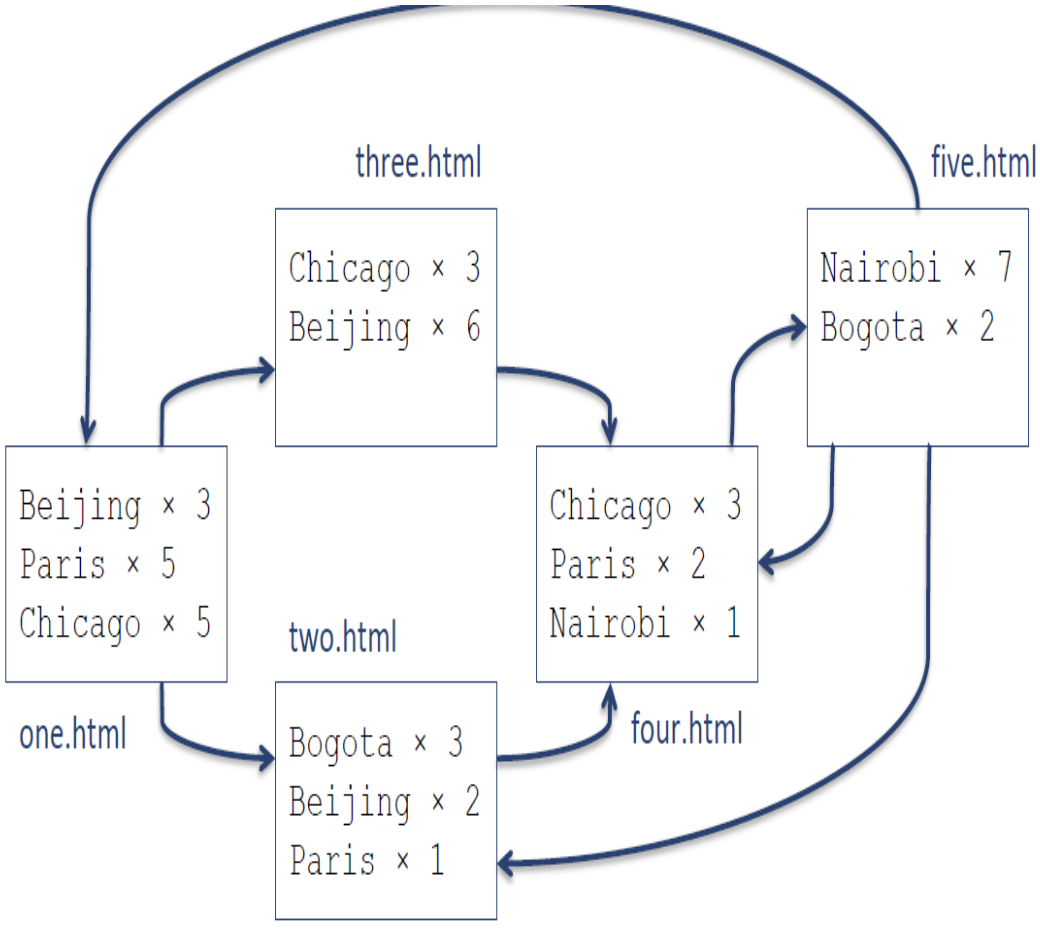

Figure 12.2(a) Database tables Hyperlink

Hyperlinks

Url	Link
one.html	two.html
one.html	three.html
two.html	four.html
three.html	four.html
four.html	five.html
five.html	one.html
five.html	two.html
five.html	four.html

Figure 12.2(b) Database tables keyword

Keywords

Url	Word	Freq
one.html	Beijing	3
one.html	Paris	5
one.html	Chicago	5
two.html	Bogota	3
two.html	Beijing	2
two.html	Paris	1
three.html	Chicago	3
three.html	Beijing	6
four.html	Chicago	3
four.html	Paris	2
four.html	Nairobi	5
five.html	Nairobi	7
five.html	Bogota	2

Practice Problem 12.4

A search engine is a server application that takes a keyword from a user and returns the URLs of web pages containing the keyword, ranked according to some criterion. In this practice problem, you are asked to develop a simple search engine that ranks web pages based on its frequency.

Write a search engine application based on the results of a web crawl that stored word frequencies in a database table keywords just like the one in Figure 12.2(b). The search engine will take a keyword, from the user and simply return the web pages containing the keyword, ranked by the frequency of the keyword, in decreasing other.

```
>>> freqSearch('lins.db')
Enter keyword: Paris
URL                FREQ
One.html           5
four.html          2
two.html           1
Enter keyword:
```

Solution Practice Problem 12.4

The search engine is a simple server program that iterates indefinitely and serves a search request in every iteration:

```
Def freqSearch(Webdb):
    '''web is a database file containing table keywords:

    freqSearch is a simple search engine that takes a keyword from the user
    and prints URLs of web pages containing it in decreasing order of frequency
    of the word'''
  con = sqlite3.connect(webdb)
  cur = con.curser()

  while True:      #serve forever
    keyword = input("Enter keyword: ")
```

```
#select web pages containing keyword in
#decreasing order of keyword frequency
Cur.execute(“””SELECT Url, Freq
                    FROM keywords
                    WHERE word =?
                     ORDER BY Freq DESC”””, (keword,))
Print(‘{:15}{:4}’ .format(‘URL’, ‘FREQ’))
For url, freq in cur:
        Print(‘{15}{:4}’.format(url, freq))
```

#####Project Assignment Starts Here!!!!!!#####

Problem 12.16

Write function ranking () that takes as input the name of a database file containing a table name Hyperlinks of the same format as the table in Figure 12.2(a). The function should add to the database a new table that contains the number of incoming hyperlinks for every URL listed in the Link column of Hyperlinks. Name the new table and its columns Ranks, URL, and Ranks, respectively. When executed against database file links, db, the wildcard query on the Rank table should produce this output:

```
>>>cur.execute (‘SELECT * FROM Ranks’)
<sqlite3.Cursor object at Ox15d2560>
>>> for record in cur:
        Print (record)

(‘five.html’,  1)
(‘four.html’,  3)
(‘one.html’,  1)
(‘three.html’,  1)
(‘two.html’,  2)
```

Problem 12.19

In Practice Problem 12.4, we develop a simple search that ranks web pages based on word frequency. There are several reasons why that is a poor way to rank web pages, including the fact that it can be easily manipulated.

Modern search engines such as Google's use hyperlink information (among other things) to rank web pages. For example, if a web page has few incoming links, it probably does not contain useful information. If, however, a web page has many incoming hyperlinks, then it likely contains useful information and should be ranked high.

Using the links, db database file obtained by crawling through the pages in Figure 12.1, and also the Rank table computed in Problem 12.16, redevelop the search engine from Practice Problem 12.4 so that it ranks web pages by number of incoming links.

```
>>>search2('links.db')
Enter keyword: Paris
URL             RANK
Four.html       3
Four.html       2
One.html        1
Enter Keyword:
```

Problem 12.24

Redo *Problem 12.16* (above) using MapReduce to compute the number of incoming links for every web page.

Problem 12.25

A web-link graph is a description of the hyperlink structure of a set of link web pages. One way to represent the web-link graph is with a list of (url, linksList) pairs with each pair corresponding to a web page; url refers to the URL of the page, and linksList is a list of URLs of hyperlinks contained in the page. Note that this information is easily collected by a web crawler.

The reverse web-link graph is another representation of the hyperlink structure of the set of web pages. It can be represented as a list of (url, incomingList) pairs with url referring to the URL of a web page and incomingList referring to a list of URLs of incoming hyperlinks. So the reverse web-link graph makes explicit incoming links rather than outgoing links. It is very useful for efficiently computing the Google PageRank of web pages.

Develop a function that takes a web-link graph, represented as described, and returns the reverse web-link graph.