INF 551 – Fall 2019

Homework #2: XML & XPath

Due: October 18, Friday (end of day)
100 points

Consider an XML document: books.xml, which stores a catalog of books. Each book has a unique id and several attributes like author and title. In this homework, you are asked first to build an inverted index and then use the index to facilitate the searching of books. Specific tasks are as follows.

1. [40 points, inverted index] Write a Python script "index.py" that takes books.xml as the input and outputs the index in an XML file: index.xml.

   You are free to choose a desirable structure for the index file. But the index should store keywords in the author, title, genre, and description attributes of books, and for each keyword, stores id's of books which contains the keyword, and for each book, indicates where (that is, in which attribute) the keyword appears in the book.

   You can assume that keywords are obtained from the content of attribute after removing white spaces and punctuation characters (except for apostrophe ')

2. [60 points, search] Write a Python script "search.py" that takes the data file (books.xml) index file (index.xml) and a string of keywords, and outputs the search result in an XML file: results.xml, which lists the documents having **all** the keywords in at least one of the attributes and also the complete content of the attributes. For example,

   python search.py books.xml index.xml "xml xslt" results.xml

   will return:

```
<results>
    <book id="bk111">
        <description>The Microsoft MSXML3 parser is covered in detail, with
                attention to XML DOM interfaces, XSLT processing, SAX and
                more.</description>
    </book>
    <book id="xxx"> … </book>
    …
</results>
```

   Note that only the *description* attribute of book "bk111" is displayed, since the book has both keywords but only in its description.

You may use library lxml for the homework. Please make sure your script compiles and runs in Python 3.

INF 551 – Fall 2019

Requirements

1. Python Environment : Python3.6
2. Packages : The Python Standard Library and lxml
3. Submission : **<firstname>_<lastname>_index.py** and **<firstname>_<lastname>_search.py**
4. Command to Execute Your Code :

```
#question 1
$ python <firstname>_<lastname>_index.py books.xml index.xml

#question 2
$ python <firstname>_<lastname>_search.py books.xml index.xml "xml xslt" results.xml
```

5. Output Format :

- For question 1, you can design your own xml structure.

But **use <keyword> as tag** for keywords and output a **valid xml document**, referring to **index.xml** in 4.

e.g. (you do not need to design exactly like this but use <keyword> tag for your inverted index)

```
<keyword>xml</keyword>
```

- For question 2, please strictly follow the output **xml format**, referring to **results.xml** in 4.

if searching for "xml xslt",

```
<results>
  <book id="bk111">
    <description>The Microsoft MSXML3 parser is covered in
      detail, with attention to XML DOM interfaces, XSLT processing,
      SAX and more.</description>
  </book>
</results>
```

if searching for "xml",

```
<results>
  <book id="bk101">
    <title>XML Developer's Guide</title>
    <description>An in-depth look at creating applications
      with XML.</description>
  </book>
  <book id="bk111">
    <description>The Microsoft MSXML3 parser is covered in
      detail, with attention to XML DOM interfaces, XSLT processing,
      SAX and more.</description>
  </book>
</results>
```

if nothing found,

```
<results/>
```

6. Execution Time : suppose to be **no more than 1 minute** for both questions.

Grading Criteria

1. If your programs can not be executed with the command specified above, there will be 40% penalty.
2. If your programs can not be executed with the required Python version, there will be 30% penalty.
3. If you use non-standard python packages (except for lxml package), there will be 30% penalty.
4. If not able to find a specified tag, such as <keyword>, then will be no point for the question.
5. If your .py takes more than 5 minutes for each to complete, there will be 20% penalty.
6. Please do not keep any "print" statements, they will lead to 10% penalty.
7. Please do not hard-code file names, else 10% penalty. Take all inputs from command line as shown.
8. Late homework will be deducted by 10% for every 24 hours that it is late. (no credit after 72 hours)