National University of Computer and Emerging Sciences, Islamabad

FAST School of Computing Spring 2022

Assignment No. 3

Important Instructions:

1. Put your .cpp and .txt files into a zip folder and upload it onto the google classroom submission folder. Name your solution file with your roll number, i.e., Assignment5_20I_1111.zip. Assignment in any other format (extension) will not be accepted and will be awarded with zero marks.
2. You are not allowed to copy solutions from other students. We will check your solution for plagiarism using plagiarism checkers. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission of your solution is not allowed. For each passing day after deadline, 20% of the marks will be deducted. Three days after the deadline, no submission will be accepted.

In this technological age, search engines have become an essential element of our life. This project will need you to create your own search engine. Your software should be able to read information from a file and search it. A console-based application is required of you. You have more flexibility in the code this time. Although we require neat and commented code, there are no constraints on the code structure.

Input Format:

A sample file is provided to you (Assignment_data.txt). In this sample file every pair of lines is a data item. The first of these lines contains an ID and a URL. The second line contains a set of keywords (space separated). There will be a blank line between pairs of lines. If you see two blank lines in a row that means you have reached the end of the file.

Your Program:

At the start, your program should ask for a filename. Then it will read the data from that file and make BST out of it. This BST will be used to search later on. Then, it should present a menu to the user. It should ask for a keyword and then output the list of URLs that match it.

Task 1:

Task 1 is all about setting up the interface for your search engine. Write a simple console-based interface. It doesn't need any functionality, but it should be able to read from a file, ask the user for keywords to search, etc. It's okay to hardcode results at this time. Your code should be able to handle invalid input.

A sample run of your program is given below. Bold lines indicate program output, lines starting with a '>' indicate user entered input.

```
Please enter a filename:
> Assignment_data.txt
File loaded successfully.
Please enter a word to search:
> life
3 result(s) found
1. http://www.gutenberg.net/dickens/otwist/4.html
2. http://www.gutenberg.net/dickens/otwist/32.html
3. http://www.gutenberg.net/dickens/otwist/40.html
```

Task 2:

Implement the search engine using a Binary Search Tree. The basic idea here is to have each node store a keyword, along with associated list of URLs. It would be helpful to implement your BST using a class, with left and right child pointers, similar to what was done in the lectures. All you have to do here is implement the insert, search, and delete operations that were explained. All these operations are required for only keywords.

BEST OF LUCK