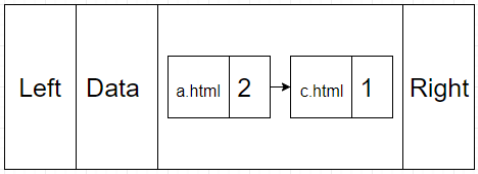
BLM19202E/ BLM22212E / SEN22212E Data Structures Project #2

MINI DESKTOP SEARCH ENGINE

(May 9, 2024, 23:59)

In this project, you are expected to develop a program in Java programming language that implements a "mini desktop search engine". The task performed by a search engine is to search through a collection of documents for specified keywords and return a list of the documents where the keywords are found.

The content of a web page is obviously correlated with the words it contains. Your task is to design and implement an algorithm that processes the documents given as input and stores the words they contain in a corresponding binary search tree and calculate the frequency of each word in each file. You are given a set of 10 html documents. Also, you are given a list of words that should be ignored during this analysis, such as common short words (a, the, …), SGML/HTML tags (lines starting with <), and punctuations (".", ",", ...). The punctuation is already separated from the words, so you do not have to worry about that. Also, you have to read one word at a time and add it to your binary search tree. The node structure for BST is as follows:



Example:

Content of the input file: *the first case is that of the wide flanged beam with convective heat transfer into the outer faces of the flanges. The second case considered is that of convective heat transfer into one side of a flat plate.*

Pre-order traversal:

first,1
case,2
beam,1
convective,2
faces,1
wide,1
flanged,1
heat,2
flanges,1
flat,1
transfer,2
into,2
outer,1
one,1
second,1
plate,1
side,1

IMPLEMENTATION DETAILS

The project will include the following requirements:

- The program should be written in Java.
- You must **design your GUI**.
- The program should read all words in the given files and build a binary search tree.
- The program should be able to search the binary search tree for a given word. The output will be the frequency of the given word in each file.
- The **post-order, pre-order, and in-order** traversal of the binary search tree for the file obtained at the end must be provided. You can use a drop-down list to select the traversal method.

All operations will be done using binary search tree and linked list. **Please don't use arrays instead, the projects prepared with array or other data structures will not be evaluated.**

You must write a report including your design and implementation details. The report also includes the output of your program.

GRADING:

1. Proper handling of capitalization and punctuation in input text (15 points).
2. Correct implementation of a binary search tree to store words and their frequencies (25 points).
3. Search the binary search tree for a given word (15 points).
4. The **post-order, pre-order, and in-order** traversal of the binary search tree (15 points)
5. GUI (10 points)
6. Report (20 points)

Notes:

By the due date, please submit the source code of your program, on the submission on LMS. And please name your files as follows: "*yournamesurnameXXX*.java", for example "BernaKirazMainClass.java".

In this assignment you must work in a group. There must be a maximum of two students in each group.

Note that projects submitted after the project's due date will not be accepted and evaluated. Please keep this in mind and **promptly start working on your projects**! Similarity test is applied with all projects. Any potential violation of this rule will lead everyone involved to failing from all projects.

Good luck ☺