Middle East Technical University

Department of Computer Engineering

CENG 351

Data Management and File Structures

Fall 2023–2024

Programming Assignment 2

Due date: 2024-01-13 23:59

1 Introduction

Your friend has recently mentioned that their infallible startup that is just like Goodreads and Spotify combined had been running into issues with their backend. You remember feeling slightly uneasy listening to their platform design. Since you are a 351 student, you know just the thing that can solve all their issues. You figure that you can implement indexing using B+ trees and become a hero. You might even get to nab that 10% share you have been promised.

In this assignment you will implement a B+ tree to store book records. The primary B+ tree will be setup using the `bookID` field. You are only required to implement certain operations, and you are supplied with a graphical user interface to check your implementation.

You are asked to complete the following functionalities;

- Add Book: Adds a new book into the tree
- Search Book: Searches for a book with the given bookID, prints the visited nodes along the path
- Print Tree: Prints the whole tree in depth first order, with proper indentation using tab characters

You will implement these three operations based on the given input and will print the necessary output. As mentioned above, you can use the GUI during your implementation, but your code should first and foremost run for the command line functionality.

2 Structure

2.1 GUI Classes

The following classes are provided to you on our ODTUClass page. Do not modify them as they will not be in your submission but will be added by us during the grading.

The GUI classes are;

- CengBook.java
- CengBookRunner.java
- CengGUI.java
- CengNodeType.java
- GUIInternalNode.java
- GUILeafNode.java
- GUILevel.java
- GUITreeNode.java
- WrapLayout.java

2.2 Implementation

You are tasked with implementing the following classes;

- CengTree.java
- CengTreeNodeInternal.java
- CengTreeNode.java
- CengTreeNodeLeaf.java
- CengTreeParser.java

The code includes `TODO` comments to guide you during the implementation.

3 Input/Output

The provided GUI is for your convenience and your homework will be graded without using it. There are also two sample input files you can load to try your implementation.

You are going to construct the primary B+ Tree using the <bookID> field. The `order` of the B+ tree will be given as an input. Note that, the `order` applies to the leaf nodes as well, a leaf node will store N records, where $d \leq N \leq 2d$.

Your implementation, while not using the GUI should listen for inputs from the command line in an infinite loop until the "quit" string is given. The other commands you should implement are listed below;

3.1 Add

- Input format:

  `add|<bookID>|<bookTitle>|<author>|<genre>`

- Example input:

  `add|99|Embrace Your Weird|Felicia Day|Nonfiction`

The pipe "|" character is used to separate different fields.
Add command does not trigger an output.

3.2 Print

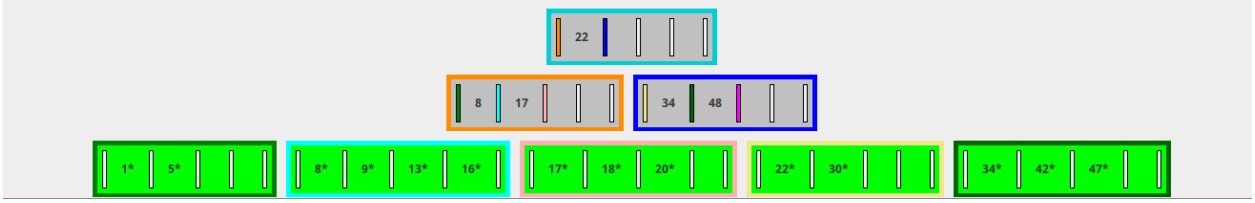
- Input format:

  `print`

- Output Format:

  Print all nodes with proper indentation (using the tab character) for each level in the tree. Non-leaf nodes use < index > and < /index > tags to print the search key. Leaf nodes print the contents between < data > and < /data > tags. Finally, records are printed between < record > and < /data > tags.

For example, take the tree below with order $d = 2$.



The below is the result of the `print` command for this tree.

```
<index>
22
</index>
        <index>
        8
        17
        </index>
                <data>
                <record>1|Bringing Down the House|Ben Mezrich|Non-fiction</record>
                <record>5|The Amateur's Mind|Jeremy Silman|Non-fiction</record>
                </data>
                <data>
                <record>8|The Box|Marc Levinson|Non-fiction</record>
                <record>9|All These Worlds|Dennis E. Taylor|Science Fiction</record>
                <record>13|The Temple of the Golden Pavilion|Yukio Mishima|Classics</record>
                <record>16|How to Take Over the World|Ryan North|Non-fiction</record>
                </data>
```

```
        <data>
        <record>17|I Could Tell You But Then You Would Have to be Destroyed by Me|Trevor Paglen|Non-fiction</record>
        <record>18|The Cloisters|Katy Hays|Mystery</record>
        <record>20|We|Yevgeny Zamyatin|Science Fiction</record>
        </data>
<index>
34
48
</index>
        <data>
        <record>22|Educated|Tara Westover|Memoir</record>
        <record>30|Hollow Kingdom|Kira Jane Buxton|Fantasy</record>
        </data>
        <data>
        <record>34|Why Fish Don't Exist|Lulu Miller|Non-fiction</record>
        <record>42|The Last Wish|Andrzej Sapkowski|Fantasy</record>
        <record>47|Dirk Gently's Holistic Detective Agency|Douglas Adams|Science Fiction</record>
        </data>
        <data>
        <record>48|Morning Star|Pierce Brown|Science Fiction</record>
        <record>51|Lost in a Good Game|Pete Etchells|Non-fiction</record>
        <record>52|Axiom's End|Lindsay  Ellis|Science Fiction</record>
        <record>58|The Humans|Matt Haig|Science Fiction</record>
        </data>
```

3.3 Search

- Input format:

  `search|<searchKey>`

- Output Format:

  Prints the visited nodes starting from root to data records in leaf nodes. Use tab characters for indentation, one character per level of the tree, while traversing down on each level in the tree. If the record is not found, Search prints "Could not find < searchKey >"

For example, using the tree given in Subsection 3.2, the `search|34` command should print;

```
<index>
22
</index>
        <index>
        34
        48
        </index>
                <record>34|Why Fish Don't Exist|Lulu Miller|Non-fiction</record>
```

`search|88`, on the other hand, should print;

```
Could not find 88.
```

4 Submission

You are expected to submit only the files given under Section 2.2. Your submission will be combined with the GUI files during compilation and execution by us. So, keep your implementation constrained only to files listed under Section 2.2. Do not use any subdirectories in your submission. Archive your submission as a `tar.gz` file with your username.

5 Notes

- Use Java version 14. This version is available on ineks as well.
- Name your submission with your username and archive it as `<e123456>.tar.gz`.
- Your submission will be extracted using `tar xvf <e123456>.tar.gz`, make sure your archive can be extracted as so.
- Grading will be done using blackbox techniques. Hence, make sure that your code runs with the following commands;
  - `javac * .java`
  - `java CengBookRunner <order> <guiOptions> <inputFileName>`.
- Make sure that your submission follows the output format presented in Section 3.
- Using implementation specific code that is not your own is strictly forbidden and constitutes as cheating. This includes but not limited to friends, previous homework, CENG homework repositories on GitHub, large language models or the Internet in general. The violators will get no grade from this assignment and will be punished according to the department regulations.