COP 5536 Fall 2017

Programming Project

Due Date: Nov 21st , 2017, 11:55 pm EST

1. General

Problem description

Develop and test a class B+Tree to store pairs of the form (key, value). For this project you are to implement a memory resident B+tree (i.e., the entire tree resides in main memory). Your implementation must be able to store multiple pairs that have the same key (i.e., duplicates). The leaves should be linked into a doubly linked list to support efficient range retrieval. The supported operations are:

1. Initialize(m): create a new order m B+Tree
2. Insert (key, value)
3. Search (key) : returns all values associated with the key
4. Search (key1,key2): returns (all key value pairs) such that key1 <= key <= key2.

Programming Environment

You may use either Java or C++ for this project. Your program will be tested using the Java or g++ compiler on the thunder.cise.ufl.edu server. So, you should verify that it compiles and runs as expected on this server, which may be accessed via the Internet.

Your submission must include a makefile that creates an executable file named treesearch.

2. Input and Output Requirements

Your program should execute using the following
$ treesearch file_name
Where file_name is the name of the file that has the input test data.

Input Format

The first line in the input file contains the order (m) of the B+Tree. Each of the remaining lines specifies a B+tree operation. The following is an example of an input file.

12
Insert(3.55,Value1)
Insert(4.01,Value10)
Insert(39.56,Value2)
Insert(-3.95,Value23)

Insert(-3.91,Value54609)
Insert(3.55,Value67)
Insert(0.02,Value98)
Search(3.55)
Search(-3.91,30.96)
Insert(3.26,Value56089)
Insert(121.56,Value1234)
Insert(-109.23,Value43234)
Search(3.71)

Output Format

For an Insert query you should not produce any output.
For a Search query you should output the results on a single line using commas to separate values. The output for each search query should be on a new line. All output should go to a file named "output_file.txt".

For a range search query, the output should be sorted according to the key (smallest to largest). Output should contain (key,value) pairs separated by commas.
If there are multiple values for some key, these may be output in any order. If a search query does not return anything you should output "Null".

The following is the output file for the above input file.

Value 67, Value1
(-3.91,Value54609), (0.02,Value98), (3.55,Value1), (3.55,Value67), (4.01, Value10)
Null

3. Submission

You must submit the following:

1.Makefile: Your make file must be directly under the zip folder. No nested directories.

2. Source Program: Provide comments.

3. REPORT:
· The report should be in PDF format.
· The report should contain your basic info: Name, UFID and UF Email account
· Present function prototypes showing the structure of your programs. Include the structure of your program.

To submit, Please compress all your files together using a zip utility and submit to the canvas system. You should look for Assignment Project for the submission.
Your submission should be named ***LastName_FirstName.zip.***

Please make sure the name you provided is the same as the same that appears on the

Canvaas system. Please do not submit directly to a TA. ***All email submission will be ignored without further notification. Please note that the due day is a hard deadline. No late submission will be allowed. Any submission after the deadline will not be accepted.***

4. Grading Policy

Grading will be based on the correctness and efficiency of algorithms. Below are some details of the grading policy.
Correct implementation and execution: 60%
Comments and readability: 15%
Report: 25%
Note: ***If you do not follow the input/output or submission requirements above, 25% of your score will be deduced.*** In addition we may ask you to demonstrate your projects.

5. Miscellaneous

- **Do not use complex data structures provided by programming languages**. You have to implement B+Tree data structure on your own using primitive data structures such as pointers. You must not use any BTree/ B+Tree related libraries.

- Your implementation should be your own. **You have to work by yourself for this assignment** (discussion is allowed). Your submission will be checked for plagiarism.