

CSCI 3287: Project 3

B+ Tree Modifications

Due Date

9:55p Fri. Dec. 2nd, 2016

No late submissions will be accepted.

Delivery Method

You will hand in a diff file that is referenced to a base implementation of a B+ Tree.

Note this is a B+ Tree, not the B Tree discussed in lectures.

Because of the nature the assignment, you should not discuss your solution in any way with your peers.

Description

This project will make modifications to an existing B+ Tree implementation. You can clone the code base from: <https://github.com/fwmler/bpt> When you are finished, you will create a diff file between your solution and the original source tree. This diff file is what you will submit to Moodle.

The code is written in straight C and there is a simple Makefile included. If you are not familiar with C, you will need to tutor yourself to be able to read this codebase.

This code base has a Command Line Interface (CLI) built into it. The CLI is scriptable and we will use test scripts to test your code. A sample test script will be published on Moodle.

This project will use interview grading. 40% of your grade will be based on the submitted materials. The remaining 60% will come from the interview portion. You must schedule an interview slot with a grader. If you do not schedule or attend a slot the interview portion will be scored zero.

Procedure

You are to change the code such that keys are maintained in the data structure in descending order rather than ascending order. Note you are modifying the order that things are sorted in, not the max number of keys per node (which is referred to the order of the tree).

1. Understand the code base

Look at the global data structures: `struct record` and `struct node`. The important functions form three groups: utility, insertion, and deletion. Your changes should not affect the operation of the program, i.e. all CLI commands should work as before. The tree will just be maintained and output in a different sorted order.

You should also be able to specify the tree order number (i.e. the max number of keys per node) the same way as before your changes.

2. *Plan your modifications*

Once you have an idea about how the code works, plan the changes you want to make.

3. *Execute the modifications in a manageable sequence*

If you have a set of modifications you want to make, try to do them in a logical order. I would also recommend adding, compiling and testing your changes incrementally.

4. *Create and submit your diff file*

The easiest thing to do here is to cd to the root of your repository and issue the git diff command and vector the output to a file, e.g.

```
% git diff > test
```

This will dump the output of git diff into a file called test that you can submit to Moodle.

5. *Be able to discuss all the changes you made to the code based on your diff and why*

The intent of the assignment is to understand and modify appropriately an existing code base that implements an important indexing data structure. Also, if you are not familiar with any of the tools discussed in this assignment, e.g. git, diff, you are responsible for coming up to speed on them.

The best student solution(s)¹ may be published as samples.

¹ As completely subjectively judged by me ;-)