COSC 6340—Database Systems

Second Project: B+ Trees

(Now due Thursday, May 7, 2015 at 11:59:59 PM)

Objective

This project will give you an opportunity to familiarize with the most commonly used data organization, B+ trees.

The Problem

You are given a list of between 400 and 500 student names containing a significant number of duplicates. As these names will not be case sensitive Doe, John, \ doe, john and DOE, JOHN will represent the same student. You might therefore want to store your keys in some canonical form, such as all lower case ore all upper case. No name will ever occupy more than 32 characters. To each distinct student name, we want to attach 224 bytes of confidential information that will initially contain 224 blank characters. Hence we can fit four student records in a one kilobyte block.

You are to write a program using a B+ tree organization to manage these data. Given that that each pointer occupies 64 bits, each B+ tree node will contain 10 keys and 11 pointers. Your program should be able:

1. To INSERT new student records into a possibly empty table;
2. To SEARCH for the record of a given student and print it out (name and confidential information);
3. To LIST all students names in the appropriate order.
4. To UPDATE a given student record;
5. To DELETE a given student record.

Your INSERT procedure should be able to detect duplicate names, reject them and write an error message. Your SEARCH and DELETE procedures should act in the same way when asked to access or delete records that do not exist.

To get full credit, your LIST procedure should be implemented efficiently using the 11^{th} pointer of the leave nodes of your B+ tree.

To simplify your life and make the assignment more interesting, your DELETE procedure should not merge B+ tree nodes when one or more records they point to are deleted. You should wait instead until the node is empty then delete it.

The Input Data

Your input data will consist of a list of record specifications separated by keywords as in:

```
*INSERT
Walker, Chris
Toleti, Srinivas
WALKER, CHRIS
Rawlinson, Edward M
Mason, Eric J
Wu, Chang-Yu
*SNAPSHOT
*SEARCH
rawlinson, edward m
Wong, Qun
*INSERT
Chiang, Mark
*DELETE
Long, Robert
*SNAPSHOT
*UPDATE
WALKER, CHRIS
New project.
```

Note that the information to be inserted in the confidential information field of a student record only appear in input lines describing updates and will always start on a new line.

All keywords will always start in column 2 and be preceded by an asterisk in column 1. Every time you program encounters the keyword SNAPSHOT, it should print out a summary of the current structure of the current index tree. This output should include:

1. The number of records in your table;
2. The number of blocks occupied by these data, B+ tree nodes excluded;
3. The depth of your B+ tree;
4. The first and last key of ***all*** your B+ tree nodes.

You will soon be provided with a sample data file available from the course page.

This handout was updated last on Thursday, April 30, 2015. Please check the course web site for updates.