

# COSC2430 Hw4: B-Trees

Created by Kunpeng Zhang (kzhang21@uh.edu)

## 1. Introduction

You will create a C++ program that can build a B-Tree. A list of integers will be given. You need to use the integers to build a B-Tree and traverse or retrieve the data required by the commands. Step by step build your B-Tree, refer to here.

<https://www.cs.usfca.edu/~galles/visualization/BTree.html>

## 2. Input and Output

The inputs are regular text file, where each line is terminated with an '\n' character.

1. There are two files will be given.
  - a) An input data file contains all the nodes of the tree. You should create the B-Tree by adding the nodes to the tree one by one.
  - b) A command file contains the commands, including "Degree=n"(n is a integer greater than or equal to 3, refer the textbook for the meaning of this "Degree") "Inorder Traversal", "Level m" (commands are case sensitive). The m in "Level m" is a number, which represents the level of the tree, the level starts from 1, which is the root of the tree.
2. Your output file contains the results of finished commands. If no result available for certain commands, you should output "empty".

All files should be processed sequentially from beginning to the end. So, the tree

should be built according to the sequence of the input values.

### 3. Input specification and explanation

The main C++ program will become the executable to be tested by the TAs. The result file should be written to another text file (output file), provided with the command line. Notice the input and output files are specified in the command line, not inside the C++ code. Notice also the quotes in the program call, to avoid Unix/Windows get confused.

Assumptions:

- For the input data file, all data are integers, no characters other than digits will be given. No empty file will be given. All integers will be separated with space characters. All integers are positive, and without sign and leading zero. When you build the B-Tree, no repeat integers appear in the B-Tree (so if a value already inserted, it should be omitted for the repeat appearance). The data may appear in more than one line. Also, you need to eliminate empty line(s).
- The total node number of the B-Tree will not exceed 1000.
- For the command file, you need to process commands one by one. One line only has one command. No empty file will be given. All command files contain the "Degree=n" command. There will be no error command, but "Level m" may request a non-existed level, for example, "Level 5" command applied to a 4-level tree, the result should be "empty" (case sensitive). There may have

empty lines between commands.

The general call to the executable is as follows:

bitree "value=input31.txt;command=command31.txt;output=output31.txt"

Call example with another command line type.

bitree value=input31.txt command=command31.txt output=output31.txt

both types may be used simultaneously.

### **Example 1 of input and output**

**input31.txt**

**2 1 3**

**command31.txt**

**Degree=3**

**Inorder Traversal**

**Level 2**

**Command line:**

**calculate input=input31.txt command=command31.txt output=output31.txt**

**output31.txt**

**1 2 3**

**1 3**

### **Example 2 of input and output**

**input32.txt**

**3 10 15 23 65 85 235 457 51 9 2 1**

**235 457 51 9 2 1**

**command32.txt**

**Degree=4**

**Level 3**

**Command line:**

**calculate input=input32.txt command=command32.txt output=output32.txt**

output32.txt

1 3 9 15 51 65 235 457

Example 3 of input and output

input33.txt

2 6 8 45 21 63 85 55 14 16 9 3 4 7 2 55 11 13 654 214 9

command33.txt

Degree=3

Level 20

Level 1

Level 2

Level 3

Level 4

Level 5

Command line:

calculate input=input33.txt command=command33.txt output=output33.txt

output33.txt

empty

11

6 21

3 8 14 63 214

2 4 7 9 13 16 45 55 85 654

empty

## 4. Requirements

- Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. The limit is 90% similarity. Code that is detected to be copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc) will be detected and result in "Fail" in the course and being reported to UH upper administration.

- Timeout is set to 2s (If your code doesn't finish within 2s for certain test case, the test case fails).

## 5. Turn in your homework

Homework 4 need to be turn in to our Linux server, follow the link here <http://cosc2430.coolpage.biz/homework.html>.

Make sure to create a folder under your root directory, name it hw4 (name need to be lower case). If you are late or if you want to regrade your code, you still need to submit your code to hw4 folder. In your hw4 folder, you should only upload the .cpp file(s) and .h file(s). Other irrelevant files may mess up with the compiler and lead to it fail your code.

ps. This document may have typos, if you think something illogical, please email TAs for confirmation.

The test cases for different submission will be different.

The input file names for different submission will be different.