

# COSC2430 Homework 5: Tree

## 1. Introduction

You will create a C++ program to implement the B-Tree. The purpose of this homework is to let students be familiar with the B-Tree.

## 2. Input and Output

### a. Input file

All values that will be added to the tree separated by a space, tab, or new line. Values will be positive.

### b. Command file

Degree of the tree: Degree=*value*  
Level to print: Level *value*

### c. Output file

The output is a single text file.  
Height of the tree and list of the numbers on the specified levels.  
If the level is empty, print *Empty*.

### d. Examples

#### i. Example 1

**input51.txt**

55 60 72

**command51.txt**

Degree=3

Level 2

**output51.txt**

Height=2

55 72

#### ii. Example 2

**input52.txt**

13	28	1	32	81
17	72	70	77	58
51	24	25	5	55
68	24	28	8	19
15	40	91	17	37
10	20	4	33	21

**command52.txt**

Degree=4

Level 10

Level 9

Level 2

Level 2

Level 1

Level 3

**output52.txt**

Height=3

Empty

Empty

5 19 21 33 70 77

5 19 21 33 70 77

13 28 51

1 4 8 10 15 17 20 24 25 32 37 40 55 58 68 72 81 91

**iii. Example 3**

**input53.txt**

228	72	177	9	284	1	169	263	237	63
183	98	149	232	100	54	236			

**command53.txt**

Degree=4

Level 3

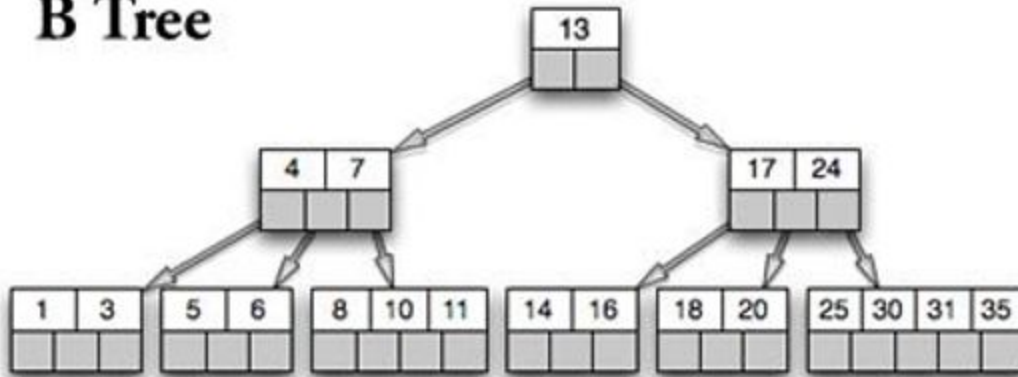
**output53.txt**

Height=3

1 22 54 63 98 100 148 149 169 183 232 236 263 284

**3. The Rules and Operations**

# B Tree



From what you learned about tree algorithms, this time you will need to construct a new type of tree, B-Tree, find its height, and perform printing by level.

Btree is an updated version of AVL tree, a self-balancing tree. B-tree stores data such that each node contains a set of values in sorted order. Each of these values has two pointers to another two child nodes. The left child node values are less than the current parent node values and the right child node values are greater than the current parent node values.

If the tree has degree 'n', then there will be maximum n-1 values and maximum n child nodes in each parent node. The command file 1st line will provide the degree for the tree. Starting from the 2nd line, each command will ask you to print the necessary data. In this Btree homework, you must ignore the duplicate values.

More info about BTree can be found in *Data Structure and Algorithms Analysis By Weiss* at <https://rizk.netlify.app/courses/cosc2430/>

## 4. Requirements

Please create the BTree manually.

The main C++ problem 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.

Homework is individual. **Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc, will be treated as copy) will be detected and result in "0" in this homework. The limit is 50% similarity. [Here](#) are some previous homework which have been found to copy each other (the main function has been deleted).**

## 5. Turn in your homework

Homework 5 needs to be turned in to our Linux server, follow the link here

[https://rizk.netlify.app/courses/cosc2430/2\\_resources/](https://rizk.netlify.app/courses/cosc2430/2_resources/)

Make sure to create a folder under your root directory, name it hw4 (name need to be lower case), only copy your code to this folder, no testcase or other files needed.

**PS:** This document may have typos, if you think something illogical, please email TAs for confirmation. Always update the newest version of the instruction from the drive every time you work on your program.