Algorithm Concepts

**CSCI-651**

**Project**

**Due Date: May 03, 2021**

## I. Project Organization

This project will study memory management.

You should do the following pieces to complete your project. Each piece is explained below:

* Design 20 points
* Code 60 points
* Output 10 points
* Conclusion 10 points

# Design

Describe your design. You can include diagrams that you have created to illustrate your design, but you should also describe it in words. You must include pseudocode for your tree insert and remove operations.

# Code

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, and should correctly implement the design. You may use Java or C++, other language choices must be approved.

# Output

Output will be graded by running your program.

## Conclusion

## Describe your project experience. What was difficult about the project? How did you approach the problem? What was learned?II. Project Description

### Parts Catalog

This project builds a parts catalog using a B-tree for storage.

**Introduction:**

Today a parts catalog would be built using a database for storage. Databases are very versatile in their ability to store and retrieve data. In this project, you will build a parts catalog system as a single-user system without the use of a database. Instead, you will create a B+-tree and store the parts data in it.

**Overview:**

This project consists of two parts: a B+-tree for storing parts data and a user interface for accessing the data. These two parts should be independent, so that changing to another storage system or another user interface is not too difficult.

Your project should allow the user to maintain a parts catalog. A flat-file of part data will be provided. Your program should begin by loading data from the flat-file into a B+-tree. Once loaded, the user can query for a particular part number, display the next 10 parts, modify the description of a part, add new parts, and delete parts. The program should end when the user chooses to exit. Upon ending, the user should be asked if changes should be saved to the flat-file.

**Details**

A B+-tree will be used for storage. While B-trees normally make use of the disk drive, for simplicity our tree will be kept in main memory. The B-tree will allow between 2 and 4 keys per index node. Each leaf node can store 16 records.

The provided flat-file will contain part data in the following format:

Columns

1-7 Part ID

16-80 Part Description

**Output**

At the end, your program should display the total number of splits, parent splits, fusions, and parent fusions. The tree depth should also be printed.

## III. Project Guidelines

### Submitting

Submit your project on Canvas. Include in your submission the following files:

1. A Word document for the written pieces of the project
2. Your source files
3. A “readme” file that describes how to compile your project.

### Resources

You are expected to write the B+-tree code yourself without consulting any other code source other than the source code in our textbook. Using existing code outside our text will be considered cheating and may result in a zero on the project. You are expected to develop your own pseudocode, though you may refer to any existing pseudocode you may find.

Any resource regarding B+-trees may be used for learning, as long as you do not use code found in such resources.

**All work is expected to be your own.**

If cheating is detected, all parties involved will be given a zero for the project and the penalty will be documented on a form that you must sign. You will be referred to the Dean’s office for further discussion. Cheating may be detected by software which compares your code with all other students’ source code as well as other code sources, or by a visual inspection of your source code.