Luke Peters

Professor Oluwayomi Adamo

CS – 300

02 December 2022

6-2 Project One

Pseudocode:

Read File:

START

Use fstream to open file

Generate method void loadCourses, string csvPath, dataStructre

Generate call to open file THEN If return value -1 THEN file not found

ELSE file found

WHILE not End of File

Read each line

IF < 2 values in line return ERROR

ELSE read parameters

IF third or more parameter

IF third or more parameter in first parameter elsewhere THEN continue

ELSE return ERROR

Close file

Vector:

Vector <Course> loadCourses, string csvPath

for int i = 0; i < file.rowCount(); i++

Generate data structure AND add to collection courses

course.courseId = file[i][1]

course.name = file[i][0]

WHILE not end of line

course.prereq. = file[i][8]

courses.push\_back(course)

HashTable:

Generate Hashtable

Generate Node struct

Unsigned int key

Vector nodes

Define tableSize

Unsigned int has int key

Generate method void HashTable AND Insert course

Generate key for given course THEN search node with key value

IF no entry found for key THEN set node to key position

ELSE IF node used THEN set old node key to Unit\_Max set key THEN set old node to course and old

Node next to null pointer

ELSE find next open node

ADD new newNode to end THEN void loadCourses, string csvPath, HashTable\* hashTable

LOOP and read rows of CSV file

For unsigned int i = 0; i < file.rowCount(); i++

Generate data structure and ADD to collection courses

course.courseId = file[i][1]

course.name = file[i][0]

WHILE not end of line

course.prereq. = file[i][8]

hashTable > Insert(course)

Tree

Define binary search tree to hold courses

BinarySearchTree\* bst

bst = new BinarySearchTree();

Generate add node method to void BinarySearchTree AND addNode(Node\* node, Course course)

IF root is null THEN add root

IF node < root THEN add to left

IF no left node THEN node becomes left

IF node > root add right

IF no right node THEN node becomes right AND void loadCourses(string csvPath, BinarySearchTree\* bst)

LOOP to read rows of CSV file

FOR unsigned int i = 0; i < file.rowCount(); i++

Generate data structure AND add to collection courses

course.courseId = file[i][1]

course.name = file[i][0]

WHILE not end of line

course.prereq. = file[i][8]

bst > Insert(course)

Vector:

Generate method to void printCourseInformation Vector <Course> courses, String courseId

OBTAIN input for courseId

WHILE vector not empty

IF input = to coursed THEN output course.courseId << output course.name

WHILE prerequisite = true THEN output course.prereq

HashTable:

Generate method to void printCourseInformation, Hashtable <Course> courses, String courseId

OBTAIN input for courseId

Set key = courseId

Set node to node.at(key)

IF present node = key

Return course, displayCourse, nodes[key].course

IF node points to null THEN return null

ELSE WHILE node not Null THEN verify against key

IF key = couseId THEN return course, displayCourse, nodes[key].course

Proceed to next node

Tree:

Generate method to void printCourseInformation, Tree <Course> courses, String courseId

Obtain input for courseId

Set present node to root

WHILE present not null

IF course.courseId = present

Return present, output course.courseId << output course.name

WHILE prerequisite = true THEN output course.prereq

IF courseIid < root

Set present to left

ELSE set present to right

Menu:

Set choice to 0;

Generate WHILE LOOP for menu

WHILE choice ≠ 4 THEN output menu choices

1. Load Course File, 2. Print Course List, 3. Print Individual Course, 4.Exit

Generate switch choice

Choice 1: loadCourses, courseFile, dataStructure

Choice 2: printSorted, courses AND call function to print

Choice 3: printCourseInformation, courseId

Choice 4: End Program

Print Sorted List:

Vector:

Generate sorted print method printSorted(courses)

Generate partition method int partition, vector <Course>& courses, int begin, int end

Set lowIndex to first element THEN set highIndex to last element

Set midpoint to lowIndex AND (highIndex - lowIndex) / 2

Set pivot to midpoint

Decrease highIndex while pivot < highIndex

Change lower values to left of pivot THEN higher values to right of pivot

Set temporary value to low index

Set low index to high index

Set high index to temporary

Generate quicksort method to void quickSort, vector <Course> AND courses, int begin, int end

Set midpoint to 0, lowIndex to being, highIndex to end

IF begin ≥ end THEN eturn

Set lowEndIndex to partition, courses, lowIndex, highIndex

Generate call to quicksort THEN quickSort, courses, lowIndex, lowEndIndex THEN quickSort, courses, lowEndIndex + 1, highIndex

Generate display course method to void displayCourse(Course course)

COUT << course.courseId << ": " << course.name << " | " << course.prereq << endl;

LOOP vector to display courses

FOR int i = 0; i < courses.size(); ++THEN displayCourse(courses[i])Tree

Generate inOrder method to void BinarySearchTree AND inOrder(Node\* node)

IF node ≠ null

Verify most left side first THEN inOrder, node > left

COUT << course.courseId << ": " << course.name << " | " << course.prereq << endl

Verify next right leaf THEN inOrder, node > right

COUT << course.courseId << ": " << course.name << " | " << course.prereq << endl

END

Table

Description automatically generatedRuntime Analysis Charts

Table

Description automatically generated

Table

Description automatically generated

In general, any program can have advantages and disadvantages of the requirements of the program. One advantage that I prefer is the HashTable. With the HashTable we can pull data from a set list in a very quick timeframe. A disadvantage in our case I would say is the use of the vector. We would like to use this method to search the specific course list. This method just takes a while to return the value we are looking for, but on the other hand it can return the data of reading a file pretty quickly.

I believe that I will plan to use the vector sort function for my final project because we can choose to use the quicksort function and we can print the whole table at once.