6/11/2023

Joey Wheeler

Project 1, Week 6

**Pseudocode Tree:**

Get file path

Open the file using file path

Check file open successful:

For each line in file:

Split the line into fields using a comma as the delimiter

If the number of fields < 2:

Print Error

End Program with Return Code 1

Extract the course number from the first field

Extract the course title from the second field

Create new course object with extracted data

For each remaining field in the line:

Extract the prerequisite course number

If the prerequisite course number does not exist in the binary tree:

Print an error message indicating a missing prerequisite course

Else:

Add the prerequisite course to the current course's list of prerequisites in the binary tree

Else:

Print Error

End Program with Return Code 1

**Pseudocode Vector:**

Initialize Vector

Open File

If File Opens Without Error

Loop Through Each Line in CSV

If Line is not Null

Parse Line Contents

If No Error Parsing

Store Contents of Line into Vector

Else

Print Error Description

Exit Program With Error Status

Else

Exit Loop

Else

Print Error Description

Exit Program With Error Status

Write Vector to File for Storage

**Pseudocode Hash:**

Get file path

Open the file using file path

Check file open successful:

For each line in file:

Split the line into fields using a comma as the delimiter

If the number of fields < 2:

Print Error

End Program with Return Code 1

Extract the course number from the first field

Extract the course title from the second field

Create new course object with extracted data

For each remaining field in the line:

Extract the prerequisite course number

If the prerequisite course number does not exist in the hash table:

Print an error message indicating a missing prerequisite course

Else:

Add the prerequisite course to the current course's list of prerequisites in the hash table

Else:

Print Error

End Program with Return Code 1

**Pseudocode Menu**

Load Data Structure: Load the file data into the data structure. Note that before you can print the course information or the sorted list of courses, you must load the data into the data structure.

Print Course List: This will print an alphanumerically ordered list of all the courses in the Computer Science department.

Print Course: This will print the course title and the prerequisites for any individual course.

Exit: This will exit you out of the program.

Display Menu with the following options and corresponding numbers:

1 - Load

2 - Print Course List

3 - Print Course

4 – Exit

Start Loop

Prompt user for selection in form of integer.

Validate user input, if invalid, display error and repeat loop

If selection == 1

Load data from CSV

repeat loop

If selection == 2

Loop through course list in memory or disk.

Display each course

repeat loop

If selection == 3

Prompt user for Course ID

Search for course in memory or disk by ID

If not found

Display Error

repeat loop

Else

Display Course Information

repeat loop

If selection == 4

break loop

Exit Program

**Efficiency Analysis**

**Tree**

Reading items from CSV – O(n) n = the number of items in the list.

Checking Tree for prerequisites – O(log n) this will reflect the number of levels of the tree.

**Vector**

Reading items from CSV – O(n) n = the number of items in the list.

Checking Tree for prerequisites – O(n^2) in worst case. This is because each time traversing the vector, each item until the found item must be checked.

**Hash**

Reading items from CSV – O(n) n = the number of items in the list.

Checking Tree for prerequisites – O(1) since the hash function will usually only need to check 1 item depending on the key used. Sometimes it will take slightly longer because of an item evaluating to the same index and how it must be stored.

I would recommend using a Hash for this application.