Brett Nottmeier

Charles Griffith

CS-300

04/11/2024

Module 6 Project One

**Vector Pseudo Code:**

**Design pseudocode to define how the program opens the file, reads the data from the file, parses each line, and checks for file format errors.**

Open file for course data

If file does not open

Output “Error unable to open.”

Return

Else

Check each line of file

If line does not have two parameters

Output “Error does not have required parameters.”

Continue to the next line on file

Validate if perquisites have a matching course line

If not found

Output “Error missing perquisite course line”

Close the file

**Design pseudocode to show how to create course objects and store them in the appropriate data structure.**

Initialize an empty vector for courses

Split course data

Create a new course object from split data

Add course object to vector

**Design pseudocode that will search the data structure for a specific course and print out course information and prerequisites**

Output “Please enter a course number:”

Read user input into a variable called userCourseNumber

For each course in data structure

If course number matches userCourseNumber

Output the course information

If the course has prerequisites

Output prerequisites

For each prerequisite

Output course information

Else

Output “No prerequisites found.”

If no course number matches userCourseNumber

Output "No course with this number found.”

**Hash Table Pseudo Code:**

**Design pseudocode to define how the program opens the file, reads the data from the file, parses each line, and checks for file format errors.**

Open file for course data

If file does not open

Output “Error unable to open.”

Return

Else

Check each line of file

If line does not have two parameters

Output “Error does not have required parameters.”

Continue to the next line on file

Validate if perquisites have a matching course line

If not found

Output “Error missing perquisite course line”

Close the file

**Design pseudocode to show how to create course objects and store them in the appropriate data structure.**

Initialize an empty hash table for courses

For each line split course data

Extract course number, title, prerequisite

Create a new course object from data

Add course object to hash table using course number as key

Close file

**Design pseudocode that will print out course information and prerequisites**

Output “Please enter a course number:”

Read user input into a variable called userCourseNumber

If userCourseNumber exists in hash table

Output the course information

If the course has prerequisites

Output prerequisites

For each prerequisite

Output course information

Else

Output “No prerequisites found.”

If no course number matches userCourseNumber

Output "No course with this number found.”

**Binary Search Tree Pseudo code:**

**Design pseudocode to define how the program opens the file, reads the data from the file, parses each line, and checks for file format errors.**

Open file for course data

If file does not open

Output “Error unable to open.”

Return

Else

Check each line of file

If line does not have two parameters

Output “Error does not have required parameters.”

Continue to the next line on file

Validate if perquisites have a matching course line

If not found

Output “Error missing perquisite course line”

Close the file

**Design pseudocode to show how to create course objects and store them in the appropriate data structure.**

**Initialize an empty binary search tree to store course objects**

**Open file for course data**

**For each line split course data**

Extract course number, title, prerequisite

Create a new course object from data

Insert course object into Binary Search Tree using course number as key

Close file

**Design pseudocode that will print out course information and prerequisites.**

Output “Please enter a course number:”

Read user input into a variable called userCourseNumber

Search for the course object with userCourseNumber in the binary search tree

If course object is found

Output the course information

If the course has prerequisites

Output prerequisites

For each prerequisite

Output course information

Else

Output “No prerequisites found.”

Else

Output "No course with this number found.”

**Create a Psuedo Code for a menu.**

Display Menu

Enter one of the following options

Output “1. Load data structure”

Output “2.Print Course List”

Output “3.Print Course”

Output "9.Exit program”

While user choice does not equal 9

Display Menu

Next input equals user choice

If user choice equal to 1

Load data structure

Else If user choice equal to 2

Print course list

Else If user choice equal to 3

Print course

Else If user choice equal to 9

Exit Program

Output “Good Bye!”

Else

“Invalid choice please try again.”

**Design pseudocode that will print out the list of the courses in the Computer Science program in alphanumeric order.**

Print course list in alphanumeric order

Create a empty list called course list

For each course in data structure

Add course number to course list

Sort the list of course numbers in alphanumeric number

For each course number in course list

Output course number and course title

**Evaluate the run-time and memory of data structures that could be used to address the requirements.**

**Vector:**

|  |  |  |  |
| --- | --- | --- | --- |
| **C**ode | **L**ine cost | **T**imes of executes | **Total cost** |
| **Open file for course data** | **1** | **1** | **1** |
| **Check each line of file** | **1** | **n** | **n** |
| **Creating course objects and storing in vector** | **4** | **n** | **4n** |
| **Worst Case** |  |  | **5n+1** |
| **Runtime** |  |  | ***O(n)*** |

**Hash Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Line Cost** | **Times of executes** | **Total cost** |
| **Opening file for course data** | **1** | **1** | **1** |
| **Check each line of file** | **1** | **n** | **9n** |
| **Creating course objects and storing in hash table** | **9** | **n** | **9n** |
| **Printing course information and prereq** | **1** | **1** | **1** |
| **Worst Case** |  |  | **10n+2** |
| **Runtime** |  |  | ***O(n)*** |

**Binary Search Tree:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Line cost** | **Times of executes** | **Total Cost** |
| **Opening file for course data** | **1** | **1** | **1** |
| **Checking each line of file** | **1** | **n** | **n** |
| **Creating course objects and storing in binary search tree** | **8** | **n** | **8n** |
| **Searching or course and printing information** | **1** | **1** | **1** |
| **Worst Case** |  |  | **9n+1** |
| **Runtime** |  |  | ***O(n)*** |

**Explain the advantages and disadvantages of each structure in your evaluation.**

**Each data structure has its advantages and weaknesses. Vectors are simple to implement, but their weakness is that they can require a longer look-up time due to having to go through the entire vector to find what it is looking for. Hash tables differ in that they are looking for a specific course due to their ability to use key-value storage, and they can remain constant when looking through a dataset no matter how large it grows. The disadvantage of a hash table is that it requires correct implementation, or it will lead to performance issues. Binary search trees allow for ordered traversal of elements, which could be helpful if ordered access to courses is needed, but maintaining balance can be challenging and may consume a lot more memory than other data structures.**

**make a recommendation for which data structure you will plan to use in your code.**

The recommendation for the data structure is a hash table. It offers efficient searching for courses and is flexible in handling data. Its look-up time remains constant no matter how large the course numbers will grow and would be well suited for this case.

Work Cited:

GeeksforGeeks. (2023, March 28). *Applications, advantages and disadvantages of hash data structure*. GeeksforGeeks. https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-hash-data-structure/

GeeksforGeeks. (2024, February 27). *Applications, advantages and disadvantages of binary search tree*. GeeksforGeeks. https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-search-tree/