Latrelle dickey

CS-300

10 August 2024

**Project one**

**Pseudocode**

Open file

Create void load for courses

Declare course identifiers and prerequisites

Create Vectors for each course

Print course information

Create a vector for each prerequisite

Print prerequisite information

If a course has a prerequisite

Print prerequisite information

For all courses If the course is the same as the course number

Print course information

If there is prerequisite information

Print prerequisite information

Open file;

Create hash table;

Create node structure;

Define nodes (Course and tablesize);

Check for keys and nodes;

If no key is found,

assign it to a node;

else if node is found;

assign to UNIT\_Max ;

else add to a new node;

Load and loop courses and CSV file;

Data structure is created for course information and prerequisites;

Course information and prerequisites is added to the hash table;

Load file

Open file

If file doesn't open

Then print ‘file not found’

Else proceed to run the program

Create course struct

Define course identifiers

Hash table is inputted

Hash table is used to create nodes

Course object is created and looped into hash table

Binary tree node is created with the courses and looped into the program

Menu is loaded

Case 1

Prompt for input

Load local course list

A list of courses is printed

Case 2

Prompt for input

Search user course list

Print searched list

Case 3

Prompt for input

Search for course information

Print individual courses

Case 4

Exit program

Print ‘Good bye’

**Program Run Time**

Vector String

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **for all courses** | 1 | n | n |
| **if the course is the same as courseNumber** | 1 | n | n |
| **for each prerequisite of the course** | 2 | 1 | 1 |
| **for each prerequisite of the course** | 1 | n | n |
| **print the prerequisite course information** | 2 | n | n |
| **Total Cost** | | | 5n + 1 |
| **Runtime** | | | O(n) |

Binary Tree

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **for all courses** | 2 | n | n |
| **if the course is the same as courseNumber** | 1 | n | n |
| **for each prerequisite of the course** | 1 | 1 | 1 |
| **for each prerequisite of the course** | 2 | n | n |
| **print the prerequisite course information** | 4 | n | n |
| **Total Cost** | | | 9n + 1 |
| **Runtime** | | | O(n) |

Hash Tables

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **for all courses** | 1 | n | n |
| **if the course is the same as courseNumber** | 1 | n | n |
| **for each prerequisite of the course** | 2 | 1 | 1 |
| **for each prerequisite of the course** | 1 | n | n |
| **print the prerequisite course information** | 4 | n | n |
| **Total Cost** | | | 7n + 1 |
| **Runtime** | | | O(n) |

**Evaluation**

The data structure I plan to use for this project is vector string. Some advantages of using the vector string are its flexibility in adjusting the size of the vector and its simplicity. Some disadvantages are the amount of memory needed for the program and its speed. Some advantages of the binary tree are the fast performance of searching, sorting, insertion, and deletion within the tree and its memory efficiency. One disadvantage of the binary tree is the space efficiency due to the limited number of nodes within the structure. Some advantages of hash tables are high-speed data retrieval and its flexibility in storing different types of data. One disadvantage of the hash tables is that it is limited to null values and capacity.