Carlos Rodriguez

Professor Yurik

CS 300

December 4, 2024

Project One

3. Print out the courses in Alphanumerical order.

Function to print the courses using the vector technique:

Sort it based off Course Number

Loop through each course:

Print the Course Number and Course Title

End Function

Function to print the courses using the hash table technique:

Create the list of keys from the hash table

Sort it alphanumerically

For each key:

Retrieve the course key from the Hash Table

Print the Course Number and Course Title

End Function  
  
Function to print the courses using the Binary Search Tree technique:

If the node is equal to NULL:

Return

Call to print the courses of the node left child

Print the node Course Number and node Course Title

Call to print the courses of the node right child

End Function

4. Run time Evaluations

Vector:

| **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** | 1 | 1 | 1 |
| **for each prerequisite of the course** | 1 | n | n |
| **print the prerequisite course information** | 1 | n | n |
| **Total Cost** | | | 4n + 1 |
| **Runtime** | | | O(n) |

Hash Table:

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Retrieve the course object using its key** | 1 | 1 | 1 |
| |  | | --- | | **for each prerequisite of the course** |  |  | | --- | |  | | 1 | n | n |
| |  | | --- | | **print the prerequisite course information** |  |  | | --- | |  | | 1 | n | n |
| **Total Cost** | | | 2n + 1 |
| **Runtime** | | | O(n) |

Binary Tree Search:

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| |  | | --- | | **Search the tree for the course by key** |  |  | | --- | |  | | |  | | --- | | log(n) |  |  | | --- | |  | | 1 | |  | | --- | | log(n) |  |  | | --- | |  | |
| |  | | --- | | **for each prerequisite of the course** |  |  | | --- | |  | | 1 | n | n |
| **print the prerequisite course information** | 1 | n | n |
| **Total Cost** | | | n+log(n) |
| **Runtime** | | | O(n) |

5. Advantages and Disadvantages of Each

Vector:

The advantages of having the vector technique are that it is simple and not a complex structure to retrieve the required information with minimal implications. It runs the print function very effective, and it can be sorted using library functions to make the arrangement a lot easier. The disadvantages of the vector technique are that it has a slower search result, if the vector is not already sorted out. The scalability is not as good because the increase in database will cause it to slow down.

Hash Table:

The advantages of having a hash table technique are that it has a fast search result for the courses and prerequisites and effectively manages storage by sorting out data that is no longer in use and deleting it and minimizes the collisions. The disadvantage of having this table is that the code structure can be more complex to build and can cause collisions/ user error to arise. This technique does not sort the data order automatically in alphanumerical order, so an additional step must be implemented to meet this requirement.

Binary Tree Search:

The Binary Tree Search advantage is that it will sort the data automatically, which makes it more efficient to print out in alphanumerical order. When retrieving a search node, it will be populated relatively quick and modifying data entries will not affect the sorted order because it will adjust accordingly. The disadvantages of this technique are the increased complexity in the structure of data, and it also requires more data storage to make the modifications and cause computational overhead.

6. Recommendation

Based off the three data structures and their analysis, I would recommend using the Hash Table data structure to use in my code. The reason for this is because looking up searches is quick and efficient because of the O(1) retrieval and the scalability of how effective it is when the data begins to grow more. Using the Big O Analysis, we see that it is good for the runtime of the searches and retrieval of the courses and prerequisites, if applicable. It needs an extra step of implementation for the alphanumerical sort, but it can be structured effectively with in text comments to document the steps. It can be a bit complex, but the structure is very effective, so the trade-off is more suitable then trying to implement a Binary Search Tree