Menu and Alphabetic Sorting Pseudocode

Vector of courses named courses

Function Menu() {

While true

Get user input

If input = 1

Load data structure

Else if input = 2

Print course list

Else if input = 3

Print course

Else if input = 4

Break

}

Function alphanumericSort {

Vector vector = vectorToBeSorted

Sort vector

For item in vector

Print item attributes

}

Worst-Case Runtime Analysis

Vector

| Code | Line Cost | # Times Executes | Total Cost |
| --- | --- | --- | --- |
| vector.insert | 1 | n | n |
| Total Cost | | | n |
| Runtime | | | O(n) |

Hash Table

| Code | Line Cost | # Times Executes | Total Cost |
| --- | --- | --- | --- |
| If currentCourse is not used | 1 | n-1 | n |
| Do currentCourse = currentCourse.next until currentCourse.next == null | 1 | (½) n (n+1) | (½) n (n+1) |
| currentCourse.next = course | 1 | n | n |
| Total Cost | | | N + n + (½) n (n+1) |
| Runtime | | | O(n) |

Tree

| Code | Line Cost | # Times Executes | Total Cost |
| --- | --- | --- | --- |
| If treeRoot course id is null  treeRoot course = course | 2 | 1 | 2 |
| If course.id > searchNode.course.id | 1 | n-1 | n-1 |
| If searchNode.right is null | 1 | (½) n (n+1) | (½) n (n+1) |
| searchNode.right = New node with course | 1 | n | n |
| searchNode = searchNode.right | 1 | ((½) n (n+1)) - 1 | ((½) n (n+1)) - 1 |
| Total Cost | | | 2 + (n-1) + (((½) n (n+1)) - 1) + n + (((½) n (n+1)) - 1  ) |
| Runtime | | | O(n) |

Data Structure Advantages/Disadvantages

Adding items to vectors is fast, but finding items takes longer. They're simple to use in programs.

Binary trees are not great at adding items quickly, but they're good at finding items. They're also moderately difficult to implement.

Hash tables are the fastest at adding items, just like vectors. They're also the quickest at finding items. However, they're the hardest to use in software.

Data Structure Selection

The data structure I have selected to use in my code is hash tables. While they are the most difficult to implement, I can experiment with different hash functions to find the fastest one.