Jack Coster

8/18/24

CS 300 Project 1

Runtime Analysis:

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

1. Hash Table

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Search for course in hash table** | 1 | 1 | 1 |
| **If course is not null** | 1 | 1 | 1 |
| **for each prerequisite of the course** | 1 | 1 | 1 |
| **print the prerequisite course information** | 1 | n | n |
| **Total Cost** | | | 2n + 1 |
| **Runtime** | | | O(1) |

1. Binary Search Tree

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Find courseNumber in BST** | 1 | Log n | Log n |
| **If node is not null** | 1 | 1 | 1 |
| **for each prerequisite of the course** | 1 | 1 | 1 |
| **print the prerequisite course information** | 1 | n | n |
| **Total Cost** | | | Log n + 2 + n |
| **Runtime** | | | O(log n) |

Advantages and Disadvantages to each method:

The vector method is easy to manage and iterate and usually simple to implement. However, for larger data sets it is inefficient due to linear time complexity. The has table has very fast lookup and insertion with average time complexity, but it can suffer from collisions more often. The Binary Search Tree provides a more efficient search with O(log n) time complexity and provides ordered data. The performance can degrade however if the table is unbalanced. Based off these findings I would recommend we use the Hash Table due to its consistent time complexity for insertion and searching making it the most efficient method for managing all the course data in this application.