CS-300: Data Structures & Algorithms
6-2 Project One Submission
Emily Black
October 13, 2024

VECTOR PSEUDOCODE

```
BEGIN

INCLUDE the VECTOR data type to be able to use vector data structures

DEFINE a vector data structure that will store course objects

OPEN .csv file

READ each line

WHILE reading each line,

    PARSE data into course information

    IF there are not at least two parameters on the line

        PRINT an error message stating at least two parameters per line is required

        BREAK

    ELSE CONTINUE

    IF the prerequisite number exists as a course

        CONTINUE

    ELSE PRINT error message stating prerequisite course does not exist

        BREAK

    DEFINE a course object to store each course's information

    APPEND the course object with the parsed data

CLOSE the course information file

PROMPT the user for a course number

IF the user entered a course number that exists within the vector data structure

    PRINT the course information

    BREAK
```

```
ELSE
    PRINT error message stating that the entered course does not exist
END
```

HASH TABLE PSEUDOCODE

```
BEGIN
INCLUDE the VECTOR data type to be able to use vector data structures
DEFINE a hash table data structure that will read from a file and store the data in a hash table
OPEN .csv file
READ each line
WHILE reading each line,
    PARSE data into course information
    IF there are not at least two parameters on the line
        PRINT an error message stating at least two parameters per line is required
        BREAK
    ELSE CONTINUE
    IF the prerequisite number exists as a course
        CONTINUE
    ELSE PRINT error message stating prerequisite course does not exist
        BREAK
    DEFINE a course object to store each course's information
    APPEND the course object with the parsed data
CLOSE the course information file
```

```
PROMPT the user for a course number

IF the user entered a course number that exists within the hash table data structure

    PRINT the course information

    BREAK

ELSE

    PRINT error message stating that the entered course does not exist

END
```

BINARY SEARCH TREE PSEUDOCODE

```
BEGIN

INCLUDE the VECTOR data type to be able to use vector data structures

DEFINE a binary search tree data structure that will read from a file and store the data in a hash table

OPEN .csv file

READ each line

WHILE reading each line,

    PARSE data into course information

    IF there are not at least two parameters on the line

        PRINT an error message stating at least two parameters per line is required

        BREAK

    ELSE CONTINUE

    IF the prerequisite number exists as a course

        CONTINUE

    ELSE PRINT error message stating prerequisite course does not exist
```

```
        BREAK

    DEFINE a course object to store each course's information

    APPEND the course object with the parsed data

CLOSE the course information file

PROMPT the user for a course number

IF the user entered a course number that exists within the binary search tree data structure

    PRINT the course information

    BREAK

ELSE

    PRINT error message stating that the entered course does not exist

END
```

MENU PSEUDOCODE

```
BEGIN

DISPLAY the following menu:

    1.  Load the file data into the data structure

    2.  Print the course list

    3.  Print the course title and its corresponding prerequisites

    9.  Exit

PROMPT the user to select an option

WHILE the selected option is not 9

    IF the user selects option 1

        CALL the corresponding function that will load the data structure, such as LoadDataStructure()
    IF the user selects option 2
        CALL the corresponding function that will print the course list, such as PrintCourseList()
    IF the user selects option 3
        CALL the corresponding function that will prompt the user to enter a course ID, validate the user’s input for existing course information, and then print that course and its corresponding prerequisites, such as PrintCourse()
    IF the user selects option 9
        EXIT the program
END
```

VECTOR PRINTCOURSELIST() PSEUDOCODE

```
BEGIN
SORT courses using an alphanumerical-based sorting algorithm
FOR existing courses within the vector data structure
    PRINT course title and its corresponding prerequisites
END
```

HASH TABLE PRINTCOURSELIST() PSEUDOCODE

```
BEGIN
INITIALIZE an empty vector data structure to hold the sorted courses
FOR each key-value pair in the original course data structure
    INSERT the course information into the sorted course data structure
SORT the sorted course data structure using an alphanumerical-based sorting algorithm
FOR existing courses within the sorted course data structure
```

```
    PRINT course title and its corresponding prerequisites

END
```

BINARY SEARCH TREE PRINTCOURSELIST() PSEUDOCODE

```
BEGIN

IF node is not null

    CALL the corresponding function that will traverse the left subtree

    CALL the corresponding function that will print the course at the current node

    CALL the corresponding function that will traverse the right subtree

END
```

CS-300: Data Structures & Algorithms
6-2 Project One Submission
Emily Black
October 13, 2024

RUNTIME ANALYSIS

VECTOR ANALYSIS

Code	Line Cost	# Times Executes	Total Cost
INCLUDE the VECTOR data type to be able to use vector data structures	1	1	1
DEFINE a vector data structure that will store course objects	1	1	1
OPEN .csv file	1	1	1
READ each line	1	n	n
WHILE reading each line,	1	n	n
PARSE data into course information	1	n	n
IF there are not at least two parameters on the line	1	n	n
PRINT an error message stating at least two parameters per line is required	1	n	n
BREAK	1	n	n
ELSE	1	n	n
CONTINUE	1	n	n
IF the prerequisite number exists as a course	1	n	n
CONTINUE	1	n	n
ELSE	1	n	n
PRINT error message stating prerequisite course does not exist	1	n	n
BREAK	1	n	n
DEFINE a course object to store each course's information	1	1	1
APPEND the course object with the parsed data	1	n	n
CLOSE the course information file	1	1	1
PROMPT the user for a course number	1	n	n
IF the user entered a course number that exists within the vector data structure	1	n	n
PRINT the course information	1	n	n
BREAK	1	n	n
ELSE	1	n	n
PRINT error message stating that the entered course does not exist	1	n	n
Total Cost			20n + 5
Runtime			O(n)

CS-300: Data Structures & Algorithms
6-2 Project One Submission
Emily Black
October 13, 2024

HASH TABLE ANALYSIS

Code	Line Cost	# Times Executes	Total Cost
INCLUDE the VECTOR data type to be able to use vector data structures	1	1	1
DEFINE a hash table data structure that will read from a file and store the data in a hash table	1	1	1
OPEN .csv file	1	1	1
READ each line	1	n	n
WHILE reading each line,	1	n	n
PARSE data into course information	1	n	n
IF there are not at least two parameters on the line	1	n	n
PRINT an error message stating at least two parameters per line is required	1	n	n
BREAK	1	n	n
ELSE	1	n	n
CONTINUE	1	n	n
IF the prerequisite number exists as a course	1	n	n
CONTINUE	1	n	n
ELSE	1	n	n
PRINT error message stating prerequisite course does not exist	1	n	n
BREAK	1	n	n
DEFINE a course object to store each course's information	1	1	1
APPEND the course object with the parsed data	1	n	n
CLOSE the course information file	1	1	1
PROMPT the user for a course number	1	n	n
IF the user entered a course number that exists within the hash table data structure	1	n	n
PRINT the course information	1	n	n
BREAK	1	n	n
ELSE	1	n	n
PRINT error message stating that the entered course does not exist	1	n	n
Total Cost			20n + 5
Runtime			O(n)

CS-300: Data Structures & Algorithms
6-2 Project One Submission
Emily Black
October 13, 2024

BINARY SEARCH TREE ANALYSIS

Code	Line Cost	# Times Executes	Total Cost
INCLUDE the VECTOR data type to be able to use vector data structures	1	1	1
DEFINE a binary search tree data structure that will read from a file and store the data in a hash table	1	1	1
OPEN .csv file	1	1	1
READ each line	1	n	n
WHILE reading each line,	1	n	n
PARSE data into course information	1	n	n
IF there are not at least two parameters on the line	1	n	n
PRINT an error message stating at least two parameters per line is required	1	n	n
BREAK	1	n	n
ELSE	1	n	n
CONTINUE	1	n	n
IF the prerequisite number exists as a course	1	n	n
CONTINUE	1	n	n
ELSE	1	n	n
PRINT error message stating prerequisite course does not exist	1	n	n
DEFINE a course object to store each course's information	1	1	1
APPEND the course object with the parsed data	1	n	n
CLOSE the course information file	1	1	1
PROMPT the user for a course number	1	n	n
IF the user entered a course number that exists within the binary search tree data structure	1	n	n
PRINT the course information	1	n	n
BREAK	1	n	n
ELSE	1	n	n
PRINT error message stating that the entered course does not exist	1	n	n
Total Cost			20n + 5
Runtime			O(n)

EVALUATION

Each data structure used in this project has its own set of advantages and disadvantages:

- Vectors open/read files and create course objects rather quickly, yet they can be slow when searching for a specific course because they must search for each item sequentially.

- Hash tables can perform quick searches using keys to map specific items in a list. Yet, since they are not sorted, an additional step of sorting the courses in alphanumerical order is required, resulting in more runtime.

- Binary search trees can sort and organize quickly and efficiently, resulting in items being located easily. However, the process can potentially result in a longer runtime due to the time it takes to make modifications.

RECOMMENDATION

Based on the above advantages and disadvantages, I recommend using a binary search tree (BST) structure because it can sort, organize, and locate items (courses) quickly and efficiently, resulting in a shorter runtime overall. Also, according to GeeksforGeeks, the BST structure performs well and is better suited for smaller data sets, such as the one in this project, with fewer elements. (GeeksforGeeks, 2024)

REFERENCES

GeeksforGeeks. (2024d, July 30). *Advantages of BST over Hash Table*.
https://www.geeksforgeeks.org/advantages-of-bst-over-hash-table/