Shaun Ryan

Professor Andujar

CS-300: DSA: Analysis and Design

10/10/2021

**Hash Table Pseudocode**

**void Menu()**

INITIALIZE choice to 0

WHILE choice does not equal 4

OUTPUT menu and prompt to screen

SET choice to user input

IF choice equals 1

PROMPT for file name

SET userInput to user input

CALL LoadHashTable(userInput)

ELSE IF choice equals 2

CALL PrintCourseList()

ELSE IF choice equals 3

PROMPT for course name

SET userInput to user input

CALL PrintCourseInformation(userInput)

OUTPUT “Goodbye” to the screen

**void LoadHashTable()**

INITIALIZE holderString

INITIALIZE prerequisiteVector, courseVector, infileVector, holderVector

INITIALIZE inFileStream

OPEN dataFile

LOOP until end of datafile

SET holderString to dataFile line

INITIALIZE Course newCourse

LOOP until end of holderString

SET Course members to holderString elements

IF newCourse number or name are empty

PRINT error “Not enough parameters for course”

EXIT

CALL insert(newCourse)

FOR all prerequisites in hash table

FOR all course numbers in hash table

CHECK that prerequisite is in courses

PRINT “Course list loaded successfully”

CLOSE datafile

RETURN

**void Insert(Course newCourse)**

INITIALIZE integer key

SET key to hash of newCourse number

INITIALIZE Node newNode

CHECK if bucket at key index is empty

INSERT newNode into hash table at key index

ELSE

LOOP through probing sequence

INSERT newNode into open bucket of hash table

RETURN

**void printCourseInformation(HashTable courses, course)**

INITIALIZE integer key

INITIALIZE node pointer

SET key to hash of course

CHECK if course at key index matches course

INSERT newNode into hash table at key index

ELSE

LOOP through probing sequence

CHECK if course at key index matches course

SET node pointer to course node

CHECK if node pointer equals null

PRINT “Course not found”

RETURN

ELSE

PRINT node->course information

RETURN

**void printCourseList(HashTable courses)**

INITIALIZE integer key

INITIALIZE node pointer

LOOP through hash table buckets

IF bucket->head is not null

LOOP through linked list

PRINT course information

Runtime Analysis

| **LoadCourses()** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **INITIALIZE variables** | 6 | 1 | 6 |
| **OPEN dataFile** | 1 | 1 | 1 |
| **LOOP until end of dataFile** | 1 | n | n |
| **SET holderString to dataFile line** | 1 | n | n |
| **INITIALIZE Course newCourse** | 1 | n | n |
| **LOOP until end of holder string** | 1 | 4n | 4n |
| **SET Course members to holderString elements** | 1 | 4n | 4n |
| **IF newCourse number or name are empty** | 1 | n | n |
| **PRINT error** | 1 | 0 | 0 |
| **IF index is greater than 1** | 1 | 2n | 2n |
| **PUSH segemet to prereqVector** | 1 | 2n | 2n |
| **IF courseVector size < 2** | 1 | n | n |
| **PRINT error message** | 1 | 0 | 0 |
| **CALL insert()** | n + 5 | n | n^2 + 5n |
| **FOR all prereqs in hash table** | 1 | 2n | 2n |
| **FOR all course numbers in hash table** | 1 | n^2 | n^2 |
| **CHECK that prereq is in courses** | 1 | n^2 | n^2 |
| **PRINT completion message** | 1 | 1 | 1 |
| **CLOSE datafile** | 1 | 1 | 1 |
| **Total Cost** | | | 3n^2 + 25n + 9 |
| **Runtime** | | | O(n^2) |

| **Insert()** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **INITIALIZE key** | 1 | 1 | 1 |
| **SET key to hash of course number** | 1 | 1 | 1 |
| **INITIALIZE newNode** | 1 | 1 | 1 |
| **CHECK if bucket is empty** | 1 | 1 | 1 |
| **INSERT newNode into hash table** | 1 | 0 | 0 |
| **ELSE** | 1 | 1 | 1 |
| **LOOP through probing sequence** | 1 | n/2 | n/2 |
| **INSERT newNode into hash table** | 1 | n/2 | n/2 |
| **Total Cost** | | | n + 5 |
| **Runtime** | | | O(n) |