Karen Schick

CS 300 DSA: Analysis and Design

Prof Norman Lippincott

April 13, 2025

**Project One - BTS**

void importDataFile(BinarySearchTree<Course> &courses, String filename) {

open file with name filename

while file not at end

read line from file

split line by commas into fields

if fields is less than two

continue to next line

set courseNumber = fields[0]

set courseName = fields[1]

create new Course object with courseNumber and courseName

for each subsequent item in fields

if item is not empty

add item to temporary prerequisites list

add Course object to courses using courseNumber as key

close file

for each course in courses

for each prerequisite in course.getPrequisites()

if not validatePrequisite()

handle error

else

addPrerequisite()

}

class Course {

private:

string courseNumber

string courseTitle

vector<string> prerequisites

public:

Course(courseNumber, courseTitle){

courseNumber = courseNumber

courseName = courseName

}

void addPrerequisites(string prerequisite){

add prerequisite to prerequisites vector

}

bool validatePrerequisite(string prerequisite, BinarySearchTree <Course> &courses, &courses){

Check if prerequisite exists in BinarySearchTree

}

function searchCourse(BinarySearchTree <Course> &courses, String courseNumber){

}

function getCourseNumber{

return courseNumber

}

function getCourseName{

return courseName

}

function getPrerequisites{

return prerequisites

}

}

void searchCourse(BinarySearchTree <Course> &courses, String courseNumber) {

for all courses

if the course is the same as courseNumber

print out the course information

for each prerequisite of the course

print the prerequisite course information

}

void inOrder(Node node) {

if node is not null

call inOrder with left nod

print getCourseNumber() getCourseName()

call inOrder with right node

}

void displaySortedCourse(BinarySearchTree <Course> &courses){

call inOrder with courses.getRoot()

}

void menuCourse(){

BinarySearchTree <Course>

set choice to 0

while choice is not 4

Print menu options

Read user input into choice

switch(choice)

case 1 :

call importDataFile(course, filename)

break

case 2 :

call displaySortedCourse(courses)

break

case 3 :

prompt for courseNumber

call searchCourse(courses, courseNumber)

break

case 4 :

exit

break

default

}

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| open file with name filename | 1 | 1 | 1 |
| While file not at end | 1 | n | n |
| Read line from file | 1 | n | n |
| Split line by commas into fields | 1 | n | n |
| If fields is less than two | 1 | n | n |
| Continue to next line | 1 |  |  |
| set courseNumber = fields[0] | 1 | n | n |
| set courseName = fields[1] | 1 | n | n |
| create new Course object with courseNumber and courseName | 1 | n | n |
| for each subsequent item in fields | 1 | n | n |
| if item not empty | 1 |  |  |
| add item to temporary prerequisites list | 1 | n | n |
| add Course object to courses using courseNumber as key | 1 | Log n | Log n |
| close file | 1 | 1 | 1 |
| for each course in courses | 1 | n | n |
| for each prerequisite in course.getPrequisites() | 1 | n | n |
| if not validatePrequisite() | 1 | 1 | 1 |
| handle error | 1 | Log n | Log n |
| else | 1 | 1 | 1 |
| addPrerequisite() | 1 | Log n | Log n |
| **Total Cost** | | | 9n + 3logn +4 |
| **Runtime** | | | O(log n) |

| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| Constructor Course(courseNumber, courseTitle) | 1 | n | n |
| addPrerequesites(string prerequisite) | 1 | 1 | 1 |
| validatePrerequisite(string prerequisite, BinarySearchTree <Course> courses &courses) | 1 | Log n | Log n |
| searchCourse(BinarySearchTree<Course> courses, String courseNumber) | 1 | n | n |
| getCourseNumber() | 1 | 1 | 1 |
| getCourseName() | 1 | 1 | 1 |
| getPrerequisites() | 1 | 1 | 1 |
| **Total Cost** | | |  |
| **Runtime** | | | O(log n) |

| **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) |