Program 5 and Program 6 - Degree Plan

This program#5 and program #6 are part of a combined group programming assignment worth a total of 100 points.

We will use a double adjacency list graph to represent courses and their prerequisites:

* Each vertex represents course information
* Edges represent prerequisite relationships from a prereq to a successor
* Since it is a double adjacency list, each prerequisite relationship is two edge nodes. There is an edge node for each of these two lists:
  + For the successor, it is on a list of its prereqs
  + For the prereq, it is on a list of successors
* See cs2123p5.h

Command File

Input file stream contains multiple types of records (terminated

by EOF). Please use getToken to get the command and then use sscanf to read any associated data.

(Program 5):

COURSE szCourseId szName

This is a course to insert. If the course already exists simply replace its szName.

PREREQ szPrereqCourseId

This is a prerequisite course for the most recent COURSE. If this course doesn't already exist, insert it and set its szName to "TBD". If the insertion of the PREREQ would cause a cycle, show a message and don't insert it.

PRTONE szCourseId

Print the vertex subscript, max dist from source (this isn't set until the DOPLAN command is executed in Pgm #6, from now use 0), course ID, course name, prereqs (max of 4), and successors. See sample output. If the course doesn't exist, show a warning.

PRTALL

Print all of the courses as is done for PRTONE. See sample output.

PRTSUCC szCourseId

Print all the successors of the specified course in a depth first manner with indentation. For each course, print the course ID and course name. Also include the specified course ID in your output. See the sample output. If the course doesn't exist, show a warning.

PRTSOURCES

Print each of the courses (ID and name) of courses that are sources (i.e., have no prereq).

PRTSINKS

Print each of the courses (ID and name) of courses that are

sinks (i.e., no other courses have them as prereqs).

MAXCHAIN szCourseId

Prints the count of the number of successors in the longest chain that begins with the specified vertex. If the course doesn't exist, show a warning.

PRTLONGS szCourseID

Prints each chain that is the longest chain of courses and prerequisites beginning at the specified course. There may be many of these. The chains are ones having a length equal to the length of this course's max chain.

\*Comment

Print the comment (as should be done with the other input commands) and ignore it

(Program 6):

PLAN szCourseId

This course is included in the set of courses we want to show in a semester by

semester plan. If the course doesn't exist, show a warning.

DOPLAN

This produces and prints the semester by semester plan. (See the sample output.)

DELETE szCourseId

Deletes the course, updates the edge lists appropriately, and frees the edge nodes. As a result of this command:

o This course will not be associated with any successors. (This also means that any successors will no longer have it as a prerequisite.)

o This course will not be associated with any prereqs. (This also means that no prereqs will have this course as a successor.)

Notes

1. Initially, use the provided data to build your Course Graph. Additional test data may be provided before the due dates.
2. Recursive functions:

void **printTraversal**(Graph graph, int iVertex, int iIndent)

This is invoked due to the PRTSUCC command. printTraversal is a recursive function which prints the current vertex information (course ID, course Name) and then uses a depth first traversal to indent and print the successors.

int **maxChainLength**(Graph graph, int iVertex)

Returns the a count of the number of vertices in the longest chain that begins with the specified vertex.

int **causesCycle**(Graph graph, int iPrereqVertex, int iVertex)

Returns TRUE if the insertion of an edge containing the course and its prereq would cause a cycle. This is used by the PREREQ command to check whether the insertion of a prereq would cause a cycle prior to actually inserting that prereq. Hint: there would be a cycle if a traversal to successors of iVertex reaches iPrereqVertex.

void **printLongChains**(Graph graph, int iVertex, int pathM[], int iLevel, int iLongLength)

Prints each chain that is the the longest chain of courses and prerequisites beginning at the specified course.

Parameters:

iVertex - begins with a starting vertex from which we want to print its longest chains. On subsequent calls, this is a successor vertex.

pathM[] - an array representing the path from the original starting vertex to the current vertex

iLevel - on each recursive call of printLongChains, this increases. It is used as the subscript into pathM[]. It is also used to test whether we reached iMaxLength.

iLongLength - known longest chain length

1. Additional functions:

int **findCourse**(Graph graph, char szCourseId[])

Returns the vertex subscript for the specified course ID. Note that this function will change for pgm6 if you do the Hash Table extra credit.

1. More functions:
   * There are many more functions you need to create. Some of them will be called based on the commands. Others will be called by other functions.
2. With program #6:
   * Additional information will be provided later
   * We will change the simple array representing vertices to use a hash array. This wil change the internals of your findCourse function and your insertion processing.
   * You will complete the code corresponding to the PRTLONG command.
   * You will read information in another file that describes a degree plan. Based on the degree plan, you will show which courses should be taken by semester.
     + Everyone will have to do the CS Core.
     + Extra credit will be given to groups who also include a Concentration sequence and electives

CSCoreArray: CS1083 CS1713 CS2123 CS2233 CS3333 CS3343 CS3423 CS3443 CS3723 CS3843 CS3853 MAT1214 MAT1224

ConcentrationReq CS3753

ConcentrationElect CS3743 CS4223 CS4233 CS4373 CS4973

Electives everything else not listed

* + **Hard-coding particular courses** (or **using other data to specify the sequence beyond what I provide**) will result in a **zero** on the entire assignment.

**What to turn in?**

Hwk 5.1: printed sheet (one per group; "selfs" must also turn in the sheet). It must be given to Larry at the beginning of class on the due date. (Do not submit via BlackBoard.)

Pgm 5 and pgm 6: via upload in BlackBoard

* team leader (for groups) must turn in the following as a single zip file (named with the group or individual name):
  + all .c files
  + .h file(s)
  + Makefile
  + output
  + instruction to TA sheet (explaining who is on the team (last name, first name) and how to compile/execute your code)
* Other participants in a group:
  + instruction sheet explaining who is on the team (last name, first name) and who is submitting the code
* "selfs" (teams of 1)must turn in the following as a single zip file(named for the individual as lastFirst.zip):
  + all .c files
  + .h file(s)
  + Makefile
  + output
  + instruction to TA sheet (explaining who is on the team (last name, first name) and how to compile/execute your code)

Hwk 6.1: via upload in Blackboard (everyone) due when pgm#6 is due

* Group Evaluation form as a PDF
* Peer Evaluation form as a PDF

CSCoreArray: CS1083 CS1713 CS2123 CS2233 CS3333 CS3343 CS3423 CS3443 CS3723 CS3843 CS3853 MAT1214 MAT1224

ConcentrationReq CS3753

ConcentrationElect CS3743 CS4223 CS4233 CS4373 CS4973

Electives 5

1. How long is the longest chain of courses for the

* Security Concentration
* Software Engineering Concentration
* Data Science Concentration

2. For a specified concentration, be able to

* show its longest chain of courses
* show a possible sequence by semester (note we don't want a gap of more than 3 semesters for a prereq)

**Partial Sample Output for Pgm #5**

>> COURSE CS1083 Intro I

>> COURSE CS1713 Intro II

>> PREREQ CS1083

>> COURSE CS2123 Data Structures

>> PREREQ CS1713

>> PRTONE CS2123

Vx TE Course Name Prereqs Successors

3 0 CS2123 Data Structures CS1713 ... ... ... -

>> PRTONE CS1713

Vx TE Course Name Prereqs Successors

2 0 CS1713 Intro II CS1083 ... ... ... CS2123

>> PRTALL

All Vertices In a List

Vx TE Course Name Prereqs Successors

1 0 CS1083 Intro I - ... ... ... CS1713

2 0 CS1713 Intro II CS1083 ... ... ... CS2123

3 0 CS2123 Data Structures CS1713 ... ... ... -

>> COURSE MAT1214 Calculus I

>> COURSE MAT1224 Calculus II

>> PREREQ MAT1214

>> COURSE MAT2233 Discrete Math

>> PREREQ MAT1214

>> PREREQ CS1713

>> COURSE MAT3333 Math Found

>> PREREQ MAT1224

>> PREREQ CS1713

>> COURSE CS3343 Analysis of Algo

>> PREREQ CS2123

>> PREREQ MAT3333

>> PREREQ MAT2233

>> \*

>> \* show some chains

>> \*

>> PRTSUCC CS1083

CS1083 Intro I

CS1713 Intro II

MAT3333 Math Found

CS3343 Analysis of Algo

MAT2233 Discrete Math

CS3343 Analysis of Algo

CS2123 Data Structures

CS3343 Analysis of Algo

>> MAXCHAIN CS1083

Maximum chain for CS1083 contains 4 courses

>> PRTLONGS CS1083

Longest Chains beginning with CS1083

CS1083 CS1713 MAT3333 CS3343

CS1083 CS1713 MAT2233 CS3343

CS1083 CS1713 CS2123 CS3343

>> COURSE CS3423 Sys Pgm

>> PREREQ CS2123

>> COURSE CS3433 Princ of Security

>> PREREQ CS3423

>> COURSE CS3443 App Pgm

>> PREREQ CS2123

>> COURSE CS3843 Comp Org

>> PREREQ CS2123

>> COURSE CS3723 Pgm Lang

>> PREREQ CS3443

>> PREREQ MAT2233

>> COURSE CS3733 Operating Systems

>> PREREQ CS3423

>> PREREQ CS3443

>> PREREQ CS3843

>> PRTALL

All Vertices In a List

Vx TE Course Name Prereqs Successors

1 0 CS1083 Intro I - ... ... ... CS1713

2 0 CS1713 Intro II CS1083 ... ... ... MAT3333 MAT2233 CS2123

3 0 CS2123 Data Structures CS1713 ... ... ... CS3843 CS3443 CS3423 CS3343

4 0 MAT1214 Calculus I - ... ... ... MAT2233 MAT1224

5 0 MAT1224 Calculus II MAT1214 ... ... ... MAT3333

6 0 MAT2233 Discrete Math CS1713 MAT1214 ... ... CS3723 CS3343

7 0 MAT3333 Math Found CS1713 MAT1224 ... ... CS3343

8 0 CS3343 Analysis of Algo MAT2233 MAT3333 CS2123 ... -

9 0 CS3423 Sys Pgm CS2123 ... ... ... CS3733 CS3433

10 0 CS3433 Princ of Security CS3423 ... ... ... -

11 0 CS3443 App Pgm CS2123 ... ... ... CS3733 CS3723

12 0 CS3843 Comp Org CS2123 ... ... ... CS3733

13 0 CS3723 Pgm Lang MAT2233 CS3443 ... ... -

14 0 CS3733 Operating Systems CS3843 CS3443 CS3423 ... -

>> PRTSUCC CS1713

CS1713 Intro II

MAT3333 Math Found

CS3343 Analysis of Algo

MAT2233 Discrete Math

CS3723 Pgm Lang

CS3343 Analysis of Algo

CS2123 Data Structures

CS3843 Comp Org

CS3733 Operating Systems

CS3443 App Pgm

CS3733 Operating Systems

CS3723 Pgm Lang

CS3423 Sys Pgm

CS3733 Operating Systems

CS3433 Princ of Security

CS3343 Analysis of Algo

>> MAXCHAIN CS1713

Maximum chain for CS1713 contains 4 courses

>> PRTLONGS CS1713

Longest Chains beginning with CS1713

CS1713 CS2123 CS3843 CS3733

CS1713 CS2123 CS3443 CS3733

CS1713 CS2123 CS3443 CS3723

CS1713 CS2123 CS3423 CS3733

CS1713 CS2123 CS3423 CS3433

>> PRTSUCC MAT1214

MAT1214 Calculus I

MAT2233 Discrete Math

CS3723 Pgm Lang

CS3343 Analysis of Algo

MAT1224 Calculus II

MAT3333 Math Found

CS3343 Analysis of Algo

>> MAXCHAIN MAT1214

Maximum chain for MAT1214 contains 4 courses

>> PRTLONGS MAT1214

Longest Chains beginning with MAT1214

MAT1214 MAT1224 MAT3333 CS3343

**Program #6 Semester Plan Information**

>> PLAN CS1083

>> PLAN CS1713

>> PLAN CS2123

>> PLAN MAT2233

>> PLAN CS3733

>> PLAN MAT3333

>> PLAN CS3423

>> PLAN CS3443

>> PLAN CS3723

>> PLAN CS3843

>> PLAN CS3853

>> PLAN MAT1214

>> PLAN MAT1224

>> \*

>> \* Add the concentration - security CS3433 CS4353 CS4363

>> \*

>> PLAN CS3433

>> PLAN CS4353

>> PLAN CS4363

>> \* Get the plan

>> \*

>> DOPLAN

Semester Plan

Semester #1

CS1083 Intro I

MAT1214 Calculus I

Semester #2

CS1713 Intro II

MAT1224 Calculus II

Semester #3

MAT3333 Math Found

MAT2233 Discrete Math

CS2123 Data Structures

Semester #4

CS3843 Comp Org

CS3443 Appl Pgm

CS3423 Sys Pgm

Semester #5

CS3723 Pgm Lang

CS3853 Computer Arch

CS3733 Operating Systems

CS3433 Princ of Security

Semester #6

CS4363 Cryptography

CS4353 Unix & Net Security