

CSE225 Data Structures, PROJECT #3

Deadline: 24.12.2017, Sunday, 24:00

Demos: 25-26.12.2017, Monday-Tuesday

Exam Scheduling Using Graph Coloring Algorithm

In this project, you will write a program that schedules the exams for the given set of students.

Your program will read a data file containing a list of student names and the classes they take. You will build a graph from the given data file. The vertices of the graph will be the courses and there will be an edge between each course of a student. *You can use either adjacency matrix or adjacency list representation.*

This graph will be used in order to schedule the final exams. *The exam schedule will have the constraint that each course of a student should be on different time slots.* For this problem, you will use the Graph Coloring Algorithm defined below.

You will process the graph with the Graph Coloring Algorithm and display a schedule of final exams. The output of your program will be

- a) a representation of the graph you generated (can be viewed like adjacency matrix or adjacency list) and
- b) the final exam schedule.

The output of your program must be in the following form:

[THE GRAPH REPRESENTATION]

*(An $n \times n$ adjacency matrix of 0s and 1s **OR** a list of size n in which the i^{th} element lists the vertices adjacent to the vertex $i - n$ is the number of courses)*

Final Exam
Period 1
<course name >
<course
name > Final Exam
Period 2
<course name >
.....
Final Exam Period P
<course name >

Graph Coloring Algorithm

1. Color first vertex with first color. *(Sort the course names in increasing order and start with the smallest one)*
2. Do following for remaining $V-1$ vertices.
Consider the currently picked vertex and color it with the lowest numbered color

that has not been used on any previously colored vertices adjacent to it. (*If your surname starts with A..İ you will traverse the graph using **breadth first search** and if your surname starts with K..Y you will traverse the graph using **depth first search method!***)

If all previously used colors appear on vertices adjacent to v, assign a new color to it.

Input file format:

On each line of the input file, there will be the name of the student and the names of the courses the corresponding student takes. There will be a colon after the name of the student, and the course names will be separated with a comma, like in the following format.

<student name1> : <course name 1> , <course name 2> , ... , <course name N>
 <student name2> : <course name 1> , <course name 2> , ..., <course name K>

Example:

Ayşe Kara : MATH101, CSE100,
 MATH259 Efe Anıl : BLAW203, MATH101
 Suat Ali Barut : STAT253,
 MATH259
 Nil Girgin : MATH259, HIST111,
 STAT253

As the output, the resulting graph can be displayed using either of the following formats:

As an adjacency matrix:

	BLAW203	CSE100	HIST111	MATH101	MATH259	STAT253
BLAW203	0	0	0	1	0	0
CSE100	0	0	0	1	1	0
HIST111	0	0	0	0	1	1
MATH101	1	1	0	0	1	0
MATH259	0	1	1	1	0	1
STAT253	0	0	1	0	1	0

(The course names in the columns of the adjacency matrix need to be in sorted order)

As an adjacency list:

BLAW203 → MATH101
 CSE100 → MATH101 – MATH259
 HIST111 → MATH259– STAT253
 MATH101 → BLAW203– CSE100 – MATH259
 MATH259 → CSE100 – HIST111– MATH101 –
 STAT253 STAT253 → HIST111– MATH259

(The course names in the adjacency list need to be in sorted order)

After displaying the matrix and using the given graph coloring algorithm, the program would produce the following schedule:

BFS:

Final Exam Period 1 => BLAW203, CSE100, HIST111

Final Exam Period 2 => MATH101, STAT253

Final Exam Period 3 => MATH259

DFS:

Final Exam Period 1 => BLAW203, CSE100, HIST111

Final Exam Period 2 => MATH101, STAT253

Final Exam Period 3 => MATH259