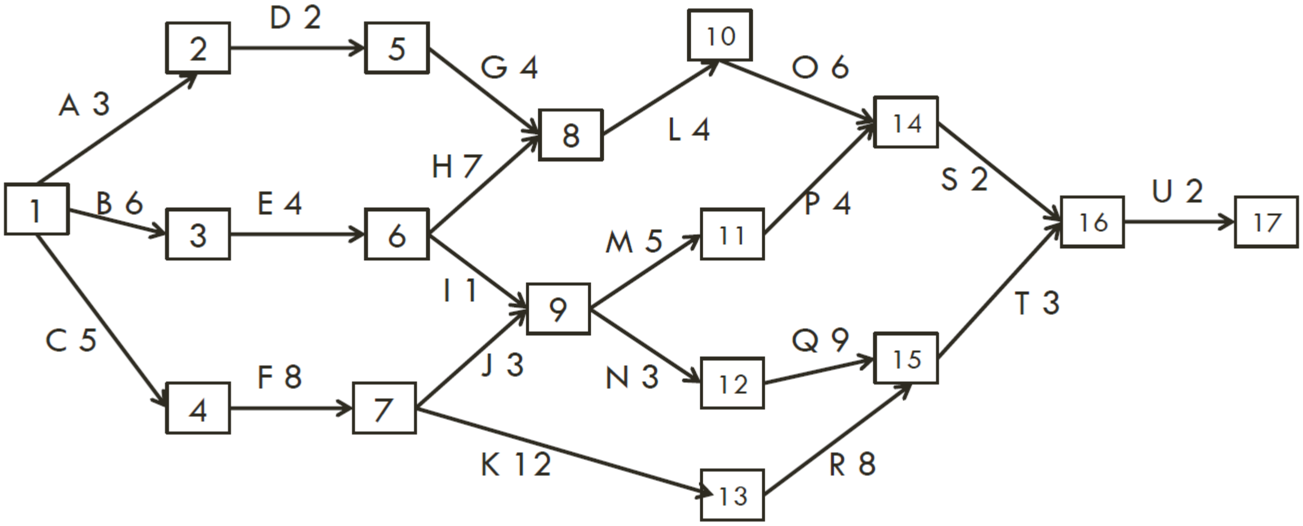
IT-309 - Assignment 7 (A7): Graph Modelling and Critical Path Analysis

Assignment Given: 11/03/2020
Assignment Due: 11/17/2020, 11:59 pm

The final two assignments, A7 and A8, are related in that they both involve graphs processing. A7 itself has two parts and is described in this document. A8 is a follow-on and its specification will be provided next week.

Part (1). This part requires you to perform analysis on a project schedule given below. It is rendered as an event node (arrow on activity) diagram. The project has 17 milestones, numbered 1 (start of project) to 17 (end of project), with the intermediate ones representing various milestones achieved during the project. A milestone could be the end of a project phase or SDLC step, the delivery of a particular product or part of a product, the end of testing, or other notable events in the project. Project tasks (activities) are ‘A’ through ‘U’ and represented on the graph edges. The numbers to the right of each task are the number of weeks required to complete it. The graph represents the task ordering (i.e. ‘A’ must be completed before ‘D’ is started).



Model the project schedule shown above as a Directed Acyclic Graph (DAG) by representing it using the three structures discussed in the lecture and chapter: adjacency list, adjacency matrix, and edge list. These can be presented using the formats shown in the slides.

Once modeled, perform a critical path analysis by determining:

- all complete paths through of the graph that originate at milestone 1 and terminate at 17
- the length of all paths
- identify the critical path and for that path calculate:
  - the earliest and latest completion date for each task
  - the earliest completion date for the entire project
  - the amount of slack time within each task

The paths should be listed as a set of activities followed by their length. For example: ADG – 9 is the path consisting of activities A, D, ang G in that order with a weighted length of 9 (which is part of a path in the above diagram). Identify which is the critical path.

The critical path data in the third bullet can be presented as a table, with each activity listed as a row. Each row has a column for earliest completion time, latest completion time, and slack. An example:

Path	**Path Length**	**Earliest Completion**	**Latest Completion**	**Slack**
A D G L O S U	23	<TBD>	<TBD>	<TBD>

Part (1) is a "paper and pencil" exercise that does not require writing or reading Python code.

Part (2). Use the Graph class provided to load information in the above project schedule into the graph. A Jupyter notebook file will be provided as a starting point. Load it and follow the instructions in reading the code and what code is needed to complete this part.

What and where to submit:

Submit the above two parts in a single Jupyter notebook. I suggest adding to the starter notebook I provide and submitting that when done. Part (1) can be submitted in one or more Markdown cells. Part (2) will involve using the code in the notebook to create the graph and perform the other required actions. You may need to provide a small amount of code in the notebook. When done upload the notebook to the A7 area in Blackboard.

How the assignment will be assessed

The code will be visually and executed. The output will be inspected and evaluated with the written report. Your program must read an input file I will provide that contains several student records, perform the hash function, and store the result in a has table.

Item	**Assessment Description**	**Max Value**
A7, Part (1)	Written text or a document containing the data structure description, paths, and critical path data incorporated into the submitted notebook. Three structures – 5 points each = 15 if correctly rendered All paths identified – up to 5 points total Accurate critical path data in tabular or other appropriate form – 5 points	25
A7, Part (2)	Project graph is correctly entered into the Graph class and the required notebook code run	15
Total		**40**