An-Najah Nation University
Faculty of Engineering

جامعة النجاح الوطنية
كلية الهندسة

Computer Engineering Department

Data Structures and Algorithms (10636211)

HW3

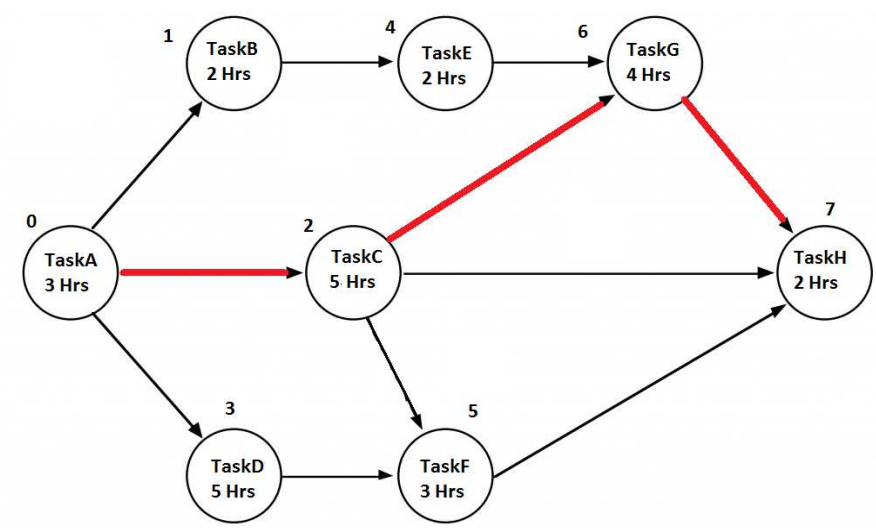
Due to 7/12/2019 ILOs [iii]

Some project planning applications use **acyclic directed graphs** (Digraphs without cycles) to represent the tasks (jobs) and task times on a project. A vertex in the graph represents a task and it stores the time the task will take. A directed edge from one vertex to another vertex represents the fact the task represented by the first vertex must be completed before the task represented by the second vertex.

Your program should be used to find the length and print the longest path (critical path) which will represent how long the entire project will take.

Each task in the project has the following valid attributes:

- TaskName [String]
- Duration [double]
- StartTime [Double]



Your program should read the required information to build the graph from a file in the following format:

- The first line gives the number of tasks (N)
- N lines, where each line gives the task-number, task-name and its duration
- One line that gives the number of edges (dependencies) (M)
- M lines, where each lines gives one edge (start-vertex and end-vertex).

```
8
0       TaskA       3
1       TaskB       2
2       TaskC       5
3       TaskD       5
4       TaskE       2
5       TaskF       3
6       TaskG       4
7       TaskH       2

11
0       1
0       2
0       3
1       4
2       6
2       7
2       5
3       5
4       6
5       7
6       7
```

The user will interact with the program using the following menu:

```
******************MAIN MENU************************
*       1- Load New File                            *
*       2- Find The Starting Task                   *
*       3- Find The Ending Task                     *
*       4- Find And Print The Critical Path         *
*       5- Exit                                     *
***************************************************
        Enter your choice:
```

Notes:

- DO NOT USE STRUCTURES. Your code should contain at least two classes:
  - Node: which is used to store the task information
  - Graph: To implement the graph and its operations as a class.
    - The constructor should takes the number of vertices as a parameter to allocate a dynamic array of pointer.
    - All the valid operations (Add Vertex, Add Edge, Compute Longest Path, and Print Longest Path) should be provided as public functions.
- You can consider the maximum number of tasks to be 20.