

Joshua Greene

Dr. Baas

Data Structures

11/4/20

## Can It Be Done? Component A

### **Description:**

This assignment makes use of a network or weighted directed graph abstract data type to represent the inter-dependencies among activities in a project. Once represented as a graph, it can be analyzed to determine if the project is feasibility, how long the project will take, and what activities are “critical” meaning that if they are not completed in time the entire project is delayed.

For this assignment, the project is represented as an Activity-On-Edge(AOE) digraph. This form of project representation may be counter-intuitive initially as the nodes of the graph represent project stages, such as Start and Complete, and the edges are used to represent project activity. The weight of an edge denotes the cost of the corresponding activity in time units. The node at the start of a directed edge is the stage that the project must be in before the activity can begin. The node at the end of the edge is the stage reached by completing the activity. Note that all incoming activities must be completed before the outgoing activities of a stage can begin. You can think of the incoming edges as pre-requisite activities for all the outgoing edges.

### **Input/Output Analysis:**

#### **Input:**

The input will be provided as follows:

5

1 2 2 6 3 9

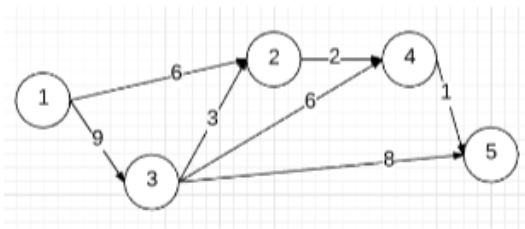
2 1 4 2

3 3 2 3 4 6 5 8

4 1 5 1

5 0

A picture of the graph created from the input example above:



The number in the first line (5) represents the number of stages (the number of vertices in the graph). Each line after that (there should be 5) contains:

1. A node number
2. The number of edges going out of this node
3. Every pair of numbers after these represent a node number and the edge cost between this node number and the node number represented by the current line.

**Output:**

Project is feasible

Ordering: 1 3 2 4 5

Stage	Early	Late
-------	-------	------

1	0	0
2	12	14
3	9	9
4	15	16
5	17	17

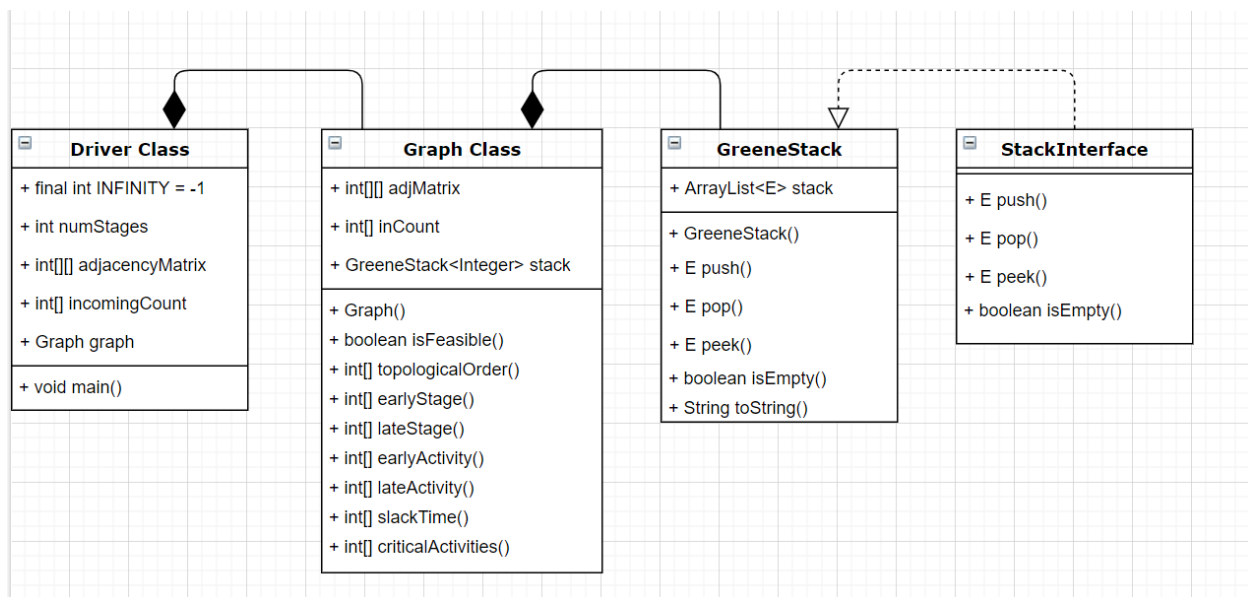
Total Project Time: 17

Activity	Early	Late
1	0	8
2	0	0
3	12	14
4	9	11
5	9	10
6	9	9
7	15	16

Critical Activities: 2, 6

First, the output displays if the project is feasible (If there are no cycles in the graph then the project will be feasible since a cycle would mean two stages depend on each other and thus make the project impossible!). It then prints the topological ordering of the stages (the nodes) so that the project can be completed. The output then prints a table containing the values for the early stage times and the late stage times. The total time to complete the project is also printed out. After all this, then another table is printed which shows the early and late stages for the activities (the edges). Finally, the critical activities (edges where the early and late times are equal!) are printed out in a comma separated list.

### UML Design:



### Design Charts:

