CS 5V81.001: Implementation of data structures and algorithms

Project 4

Akshay Thakare (ast140230)

Description: To find the critical paths in projects (longest path in DAG)

Input specification:

First line of input may have a comment (if line starts with a #). First line of the actual input has N, the number of tasks, and C,the number of precedence constraints. The next N integers are thedurations of the tasks. There may be arbitrary line breaks. Then the next C lines that follows has 2 integers, gives a precedence constraint between 2 tasks. Tasks are numbered 1..N. Do not assume any ordering of the tasks or the edges. Do not process any lines in the input file beyond this.
Limits: There are at most 1000 tasks and 6000 precedence constraints.

Terms Used:

1. Predecessors: Tasks that must be completed before any subsequent, dependent task can begin.

2. EC: Earliest completion time of the task. The earliest point in the schedule at which a task can finish.

3. LC: Latest completion time of the task. The latest point in the schedule at which a task can finish without causing a delay in the overall timeline.

4. Slack: LC – EC, Buffer for the task for its completion without causing delay in the overall timeline of the project.

Classes in the Project:

1. **Graph:** Class used to create graph, its topological order and calls method from PERT class to calculate the critical paths.

2. **PERT:** Class used to calculate EC, LC, slack, critical paths and display the tasks information(EC, LC, slack).

Results:

Input File Name	Output	Time in ms
in-l.txt	98 50 2	0 msec.
in-k.txt	323 40 2	31 msec
in-d.txt	596 55 2	31 msec
in-c.txt	183 18 2	16 msec
channel-x.txt	17 17 54	0 msec
channel-l.txt	30 28 192	32 msec
channel-d.txt	85 68 3072	578 msec