

Context:

I have a need for a Task Sorter to manage a list of tasks. Some tasks are dependent on others, which they cannot start until the pre-requisite task(s) is/are completed.

(Note: tasks can be executed simultaneously / at the same time)

(Note: A Direct Acyclic Graph (DAG), can be particularly useful)

The information about the tasks and the dependencies among the tasks are stored in a text file. The information about a task includes a task ID, which is a string, and the time needed to complete the task, which is a positive integer, and a list of tasks that the task depends on. The information about one task is stored in a separated line in the text file.

(Note: text in brackets are for description purposes and not included in the actual text file)

An example text file:

T1, 100	<i>(T1 takes 100 units of time to complete)</i>
T2, 30, T1	<i>(T2 takes 30 units of time to complete, but can't start till T1 is complete)</i>
T3, 50, T2, T5	<i>(T3 takes 50 units of time to complete, but can't start till T2 & T5 are complete)</i>
T4, 90, T1, T7	<i>(T4 takes 90 units of time to complete, but can't start till T1 & T7 are complete)</i>
T5, 70, T2, T4	<i>(T5 takes 70 units of time to complete, but can't start till T2 & T4 are complete)</i>
T6, 55, T5	<i>(T6 takes 55 units of time to complete, but can't start till T5 is complete)</i>
T7, 50	<i>(T7 takes 50 units of time to complete)</i>

Features of the Application:

1. Be in the form of a Microsoft Console Application with a command line menu allowing the user to do the following:
 - a. Ask the user to enter the name of a text file in which the information about the tasks in a project and the dependencies among the tasks are stored and read the information from the text file into the system.
 - b. Add a new task with time needed to complete the task and other tasks that the task depends on into the project.
 - c. Remove a task from the project.
 - d. Change the time needed to be completed.
 - e. Save the (updated) information about the tasks and dependencies back to the opened input text file.

Main Tasks:

1. Find the **sequence of the tasks** that does not violate any task dependency and save the sequence to a text file (named 'Sequence.txt').

For the above example, the content in Sequence.txt should look like:

Sorted Order:

T1, T7, T2, T4, T5, T3, T6

(Note: in the case of the same commencement time, the order would be in the order received)

2. Find the **earliest possible commencement time** for each of the tasks and save the solution into a text file (named 'EarliestTimes.txt').

(Note: text in brackets are for description purposes and not included in the actual text file)

(Note: the order of tasks should be the same as the original input file)

For the above example, the content in 'EarliestTimes.txt' should look like:

T1, 0	<i>(Immediate commencement, 0 units of time)</i>
T2, 100	<i>(Commence after T1 is complete, at 100 units of time)</i>
T3, 260	<i>(Commence after T2 & T5 are complete, T5 starts at 190 units + 70 units to complete, is longer than T2, which is 100 units + 30 units to complete)</i>
T4, 100	<i>(Commence after T1 & T7 are complete, which ever is longer, T1 takes 100)</i>
T5, 190	<i>(Commence after T2 & T4 are complete, T4 starts at 100 units + 90 units to complete, is longer than T2, which is 100 units + 30 units to complete)</i>
T6, 260	<i>(Commence after T5 is complete, T5 starts at 190 units + 70 units to complete)</i>
T7, 0	<i>(Immediate commencement, 0 units of time)</i>

Software Requirements:

- C# programming language.
- Using 'Console Application' of Microsoft Visual Studios 2022.
- The outputted files are saved to the 'Downloads' folder.

Additional Notes:

1. All tasks must have a way of completing.
2. Only 1 task can be added or removed at a time.

Interface:

Please enter the text file name you want to input:

Example.txt (user input)

(Note: display 'File does not exist' error message if file not found or non-text file, display 'Please enter the text file name you want to input:' again)

File loaded. (display interface menu)

Choose one of the following options:

- (1) Add a task.
- (2) Remove a task.
- (3) Update a task completion time.
- (4) Save tasks to Downloads folder. (create & save the 'Sequence.txt' & 'EarliestTimes.txt' to folder)

Case 1: Add

1 (user choose option 1)

Task number: 7 (user input)

Completion time: 60 (user input)

Any dependencies: T4 (user input) - or - Any dependencies: 0 (user input for no dependencies)

Task added.

(Update original text file of Tasks, adding this Task on a line below, return to interface menu)

(Note: display 'Task already in collection' error message if task already exist in collection)

Case 2: Remove

2 (user choose option 2, print all tasks in the collection)

T1, 100

...

T6, 55, T5

T7, 50

Enter the task to remove: T5 (user input)

Task removed.

(Update original text file of Tasks, deleting this Task, return to interface menu)

(Note: display 'Task not found' error message if task is not found in collection)

Case 3: Update completion time

3 *(user choose option 3, print all tasks in the collection)*

T1, 100

...

T6, 55, T5

T7, 50

Enter the task to update: 6 *(user input)*

Enter the new completion time: 67 *(user input)*

Task updated.

(Update original text file of Tasks, return to interface menu)

(Note: completion time must be a positive integer and >0)

(Note: display 'Enter a correct completion time' error message if negative number or 0 is entered)

Case 4: Save files

4 *(user choose option 4)*

(Note: nothing to display, 'Sequence.txt' & 'EarliestTimes.txt' created and saved to folder)