Math 140: Chapter 2

2.1 Scheduling - Getting Started

The problem of scheduling is fairly universal. Contractors need to schedule workers and subcontractors to build a house as quickly as possible. A magazine needs to schedule writers, editors, photographers, typesetters, and others so that an issue can be completed on time.

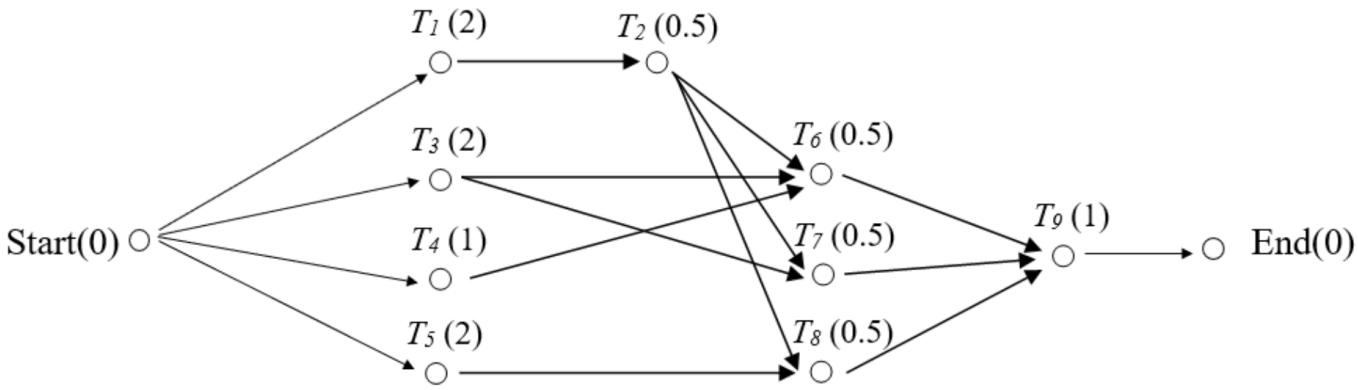
Example 1. An auto shop is converting a car from gas to electric.

- Task 1: Remove engine and gas parts (2 days)
- Task 2: Steam clean the inside of the car (0.5 day)
- Task 3: Buy an electric motor and speed controller (2 days for travel)
- Task 4: Construct the part that connects the motor to the car's transmission (1 day)
- Task 5: Construct battery racks (2 days)
- Task 6: Install the motor (0.5 day)
- Task 7: Install the speed controller (0.5 day)
- Task 8: Install the battery racks (0.5 day)
- Task 9: Wire the electricity (1 day)

We can model this problem with a digraph.

Digraph

A complete digraph for our car conversion would look like this:



Processors

Assumptions:

- Every processor can do every task
- Only one processor can work at a task at a time

One processor Solution:

Finishing time

For our auto example:
Optimal schedule

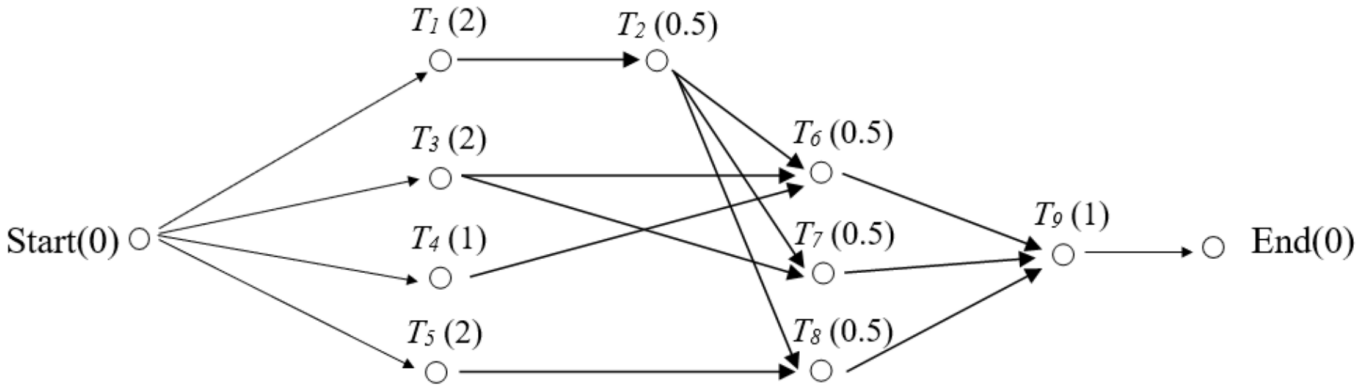
Critical Time

2.2 List Processing Algorithm - LPA

To create a more procedural approach, we might begin by somehow creating a priority list.
Priority List A priority list is a list of tasks given in the order in which we desire them to be completed.
Priority lists can be given to you or you have to create them.

Example 2. Schedule the project with two processors, using the priority list:

$$T_1, T_3, T_4, T_5, T_6, T_7, T_8, T_2, T_9$$

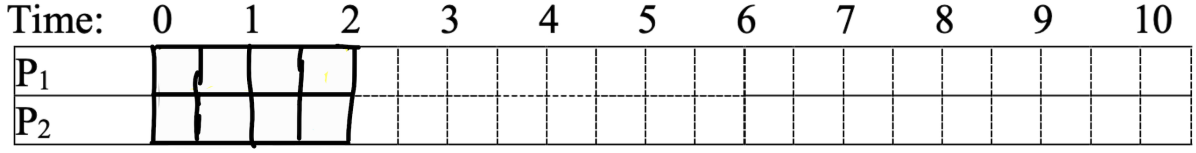


Solution for our example when we are given the priority list.

$$T_1, T_3, T_4, T_5, T_6, T_7, T_8, T_2, T_9$$

First rewrite the priority list with the amount of time each taks takes:

Next, draw a rectangle and use as many rows as you have processors. Mark completion times above first row.



Algorithm:
Time 0:

- Mark ready tasks by circling them.
- Assign tasks to processor and mark those tasks as in progress by slashing them.
- Mark the schedule up to here.

$$T_1, T_3, T_4, T_5, T_6, T_7, T_8, T_2, T_9$$

We jump to the next time when the next task completes, which is at time 2.

Next Time:
Update priority list.

- We mark those tasks as complete (by crossing them).
- Circle new available tasks
- Assign tasks to processor and mark those tasks as in progress by slashing them.
- We assign the next ready tasks on the list.
- Mark the schedule up to here.

Next Time:
Update priority list.

- We mark those tasks as complete (by crossing them).
- Circle new available tasks
- Assign tasks to processor and mark those tasks as in progress by slashing them.
- We assign the next ready tasks on the list.
- Mark the schedule up to here.

Next Time:
Update priority list.

- We mark those tasks as complete (by crossing them).
- Circle new available tasks
- Assign tasks to processor and mark those tasks as in progress by slashing them.

- We assign the next ready tasks on the list.
- Mark the schedule up to here.

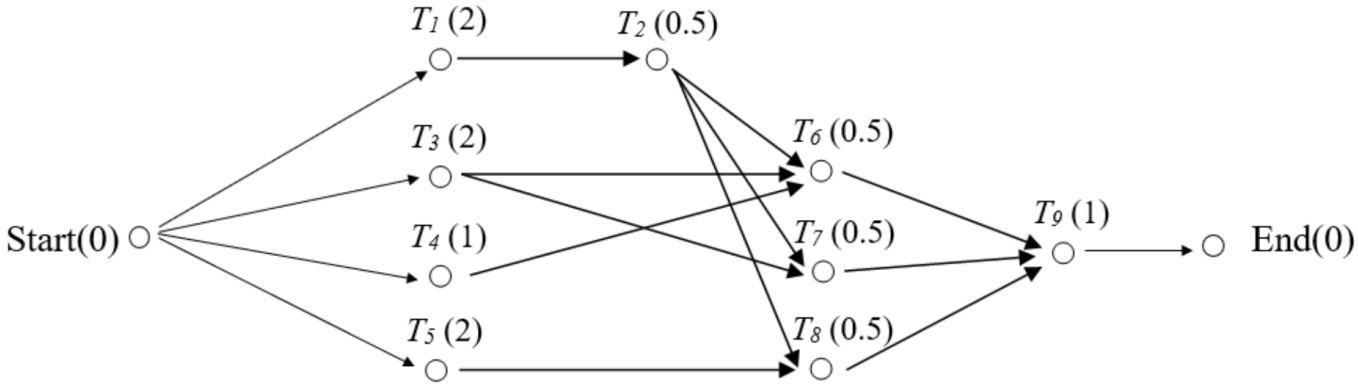
Next Time:
Update priority list.

- We mark those tasks as complete (by crossing them).
- Circle new available tasks
- Assign tasks to processor and mark those tasks as in progress by slashing them.
- We assign the next ready tasks on the list.
- Mark the schedule up to here.

Next Time:
Update priority list.

- We mark those tasks as complete (by crossing them).
- Circle new available tasks
- Assign tasks to processor and mark those tasks as in progress by slashing them.
- We assign the next ready tasks on the list.
- Mark the schedule up to here.

Example 3. How long does it take with 3 processors?



Use same priority list.

$$T_1, T_3, T_4, T_5, T_6, T_7, T_8, T_2, T_9$$