Math 140: Thursday, Sep 26, Day 10 Notes

Agenda for Thursday, Sep 26,2024

Coming up

- Office Hour **today 1-2:15pm** in C162

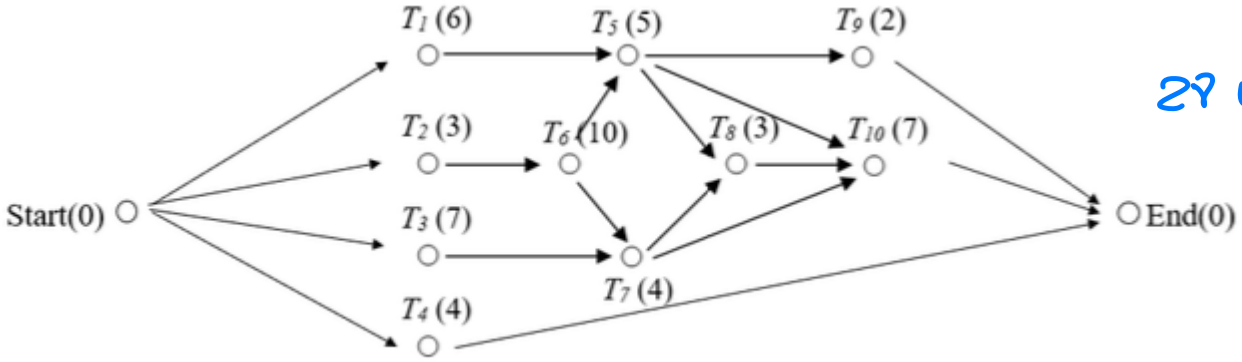
Recall: Scheduling problem

Next, we will explore two algorithms for selecting a priority list.

Decreasing Time Algorithm

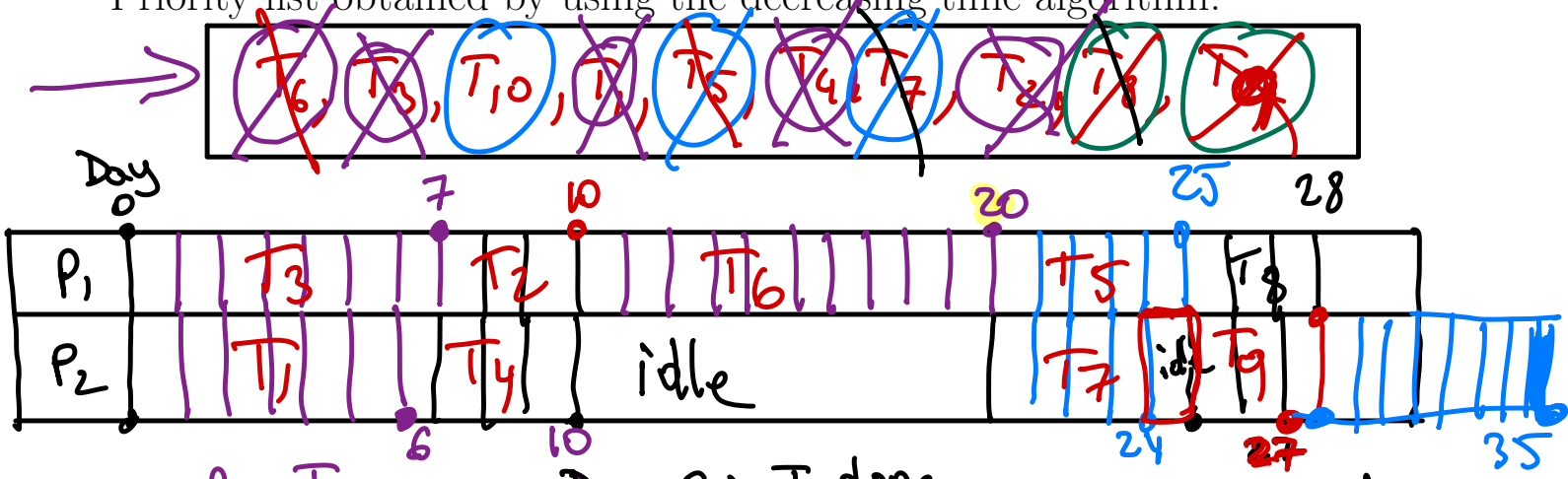
Create the priority list by listing the tasks in order from longest completion time to shortest completion time.

Example 1. Consider the scheduling problem represented by the digraph below. Create a priority list using the decreasing time list algorithm, then use it to schedule for two processors using the list processing algorithm.



29 days

Priority list obtained by using the decreasing time algorithm:



Day 0: $P_1 \to T_3$, $P_2 \to T_1$

Day 6: T_1 done, $P_2 \to T_4$

Day 7: T_3 done, $P_1 \to T_2$

Day 10: T_2, T_4 done, $P_1 \to T_6$, $P_2 \to$ idle

Day 20: T_6 done, $P_1 \to T_5$, $P_2 \to T_7$

Day 24: T_7 done, $P_2 \to$ idle

Day 25: T_5 done, $P_1 \to T_8$, $P_2 \to T_9$

Day 27: T_9 done, P_2 idle

Day 28: T_8 done, $P_2 \to T_{10}$

The **critical path algorithm** allows you to create a priority list based on idea of critical paths.

Critical path: dictates the minimum completion time for a schedule.

Example 2. What is the critical path for the previous example?

$T_2 \to T_6 \to T_5 \to T_8 \to T_{10}$

28 days

Critical Path Algorithm for building a priority list

1. Find the critical path.
2. The first task in the critical path gets added to the priority list.
3. Remove that task from the digraph
4. Repeat, finding the new critical path with the revised digraph.

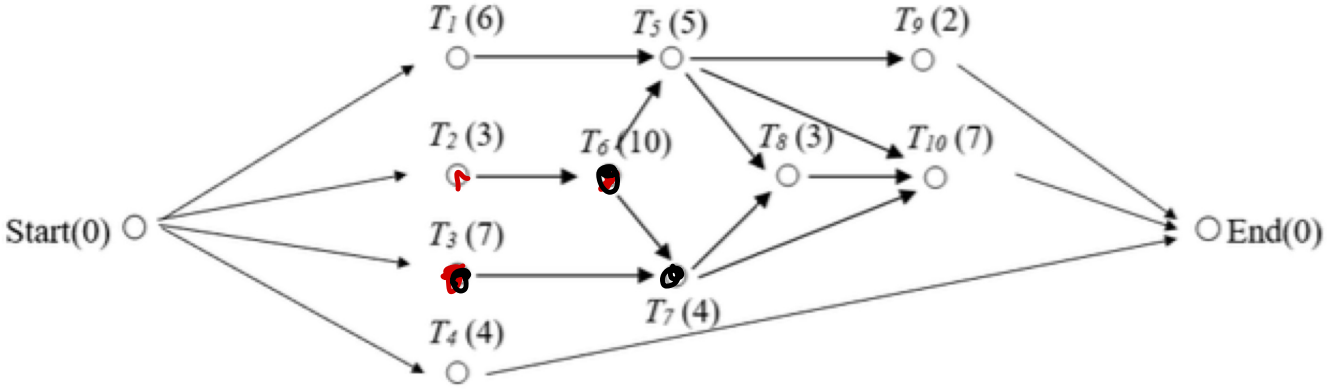
Example 3. Use the above algorithm to find the priority list for the same digraph.

Priority list:

$T_2, T_6, T_1, T_3, T_5, T_7, T_8, T_{10}, T_4, T_9$

- 2nd critical path T_6, T_5, T_8, T_{10}
- 3rd c-path T_1, T_5, T_8, T_{10}
- T_5, T_7, T_8, T_{10}

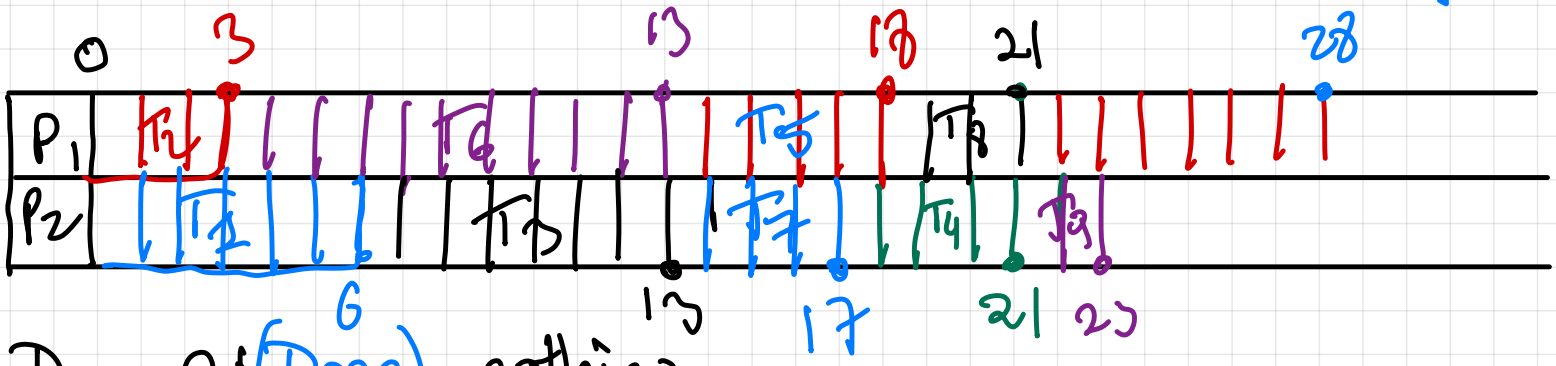
algorithm.



Priority list:

T_2, T_6, T_1, T_3, T_5, T_7, T_8, T_{10}, T_4, T_9

completion time!



Day 0: (Done) nothing
(Available) circle available tasks
(Assign) $P_1 \to T_2$ & $P_2 \to T_1$

Day 3: T_2 done
T_6 available
$P_1 \to T_6$

Day 6: T_1 done
no new available
$P_2 \to T_3$

Day 13: T_6 & T_3 done
T_5, T_5 avai
$P_1 \to T_5$
$P_2 \to T_7$

Day 17: T_7 done
- nothing new
- $P_2 \rightarrow T_4$

Day 18: T_5 done
- T_8 & T_9 become available
- $P_1 \rightarrow T_8$

Day 21 T_4 & T_3 done
- T_{10} becomes available
- $P_1 \rightarrow T_{10}$ (or P_2 ; it doesn't matter)
- $P_2 \rightarrow T_9$

The **critical path algorithm** allows you to create a priority list based on idea of critical paths.
Critical path:

Example 2. What is the critical path for the previous example?

Critical Path Algorithm for building a priority list

1. Find the critical path.
2. The first task in the critical path gets added to the priority list.
3. Remove that task from the digraph
4. Repeat, finding the new critical path with the revised digraph.

Example 3. Use the above algorithm to find the priority list for the same digraph.