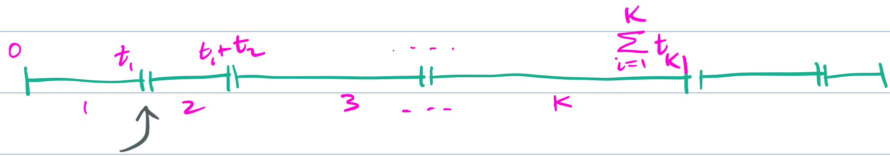
Greedy Algs II

Sept 16, 2025

Another Scheduling Problem:

- 1 processor
- n jobs, I_i, $1 \le i \le n$ $\quad$ I_i has deadline d_i and execution time t_i

Optimal Schedule := a permutation of the jobs such that max lateness of completion is MINIMIZED.



WLOG, we can assume that there is no gap between ending one job and starting another.

$\$$ $I_1, I_2, \ldots, I_n$ is the schedule we land on.

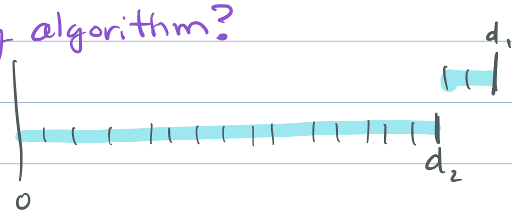
f_k = finish time of $I_k = \sum_{i=1}^{k} t_i$

ℓ_k = lateness of $I_k = \max(0, f_k - d_k)$

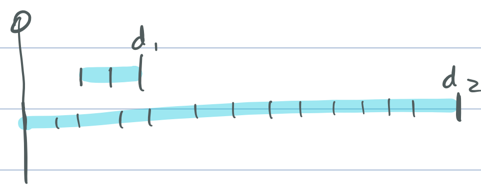
We want to minimize the maximum lateness over all the jobs.

What would be a good greedy algorithm?

- "Shortest job first"



- "Least slack time first" (slack $= d_i - t_i$)



"Earliest Deadline First" = sort the jobs by d_i
(EDF) and execute in that order.

Theorem: EDF is optimal for minimizing max latency.

First, let us acknowledge that there is no use for idle time for the processor throughout the execution of any schedule.

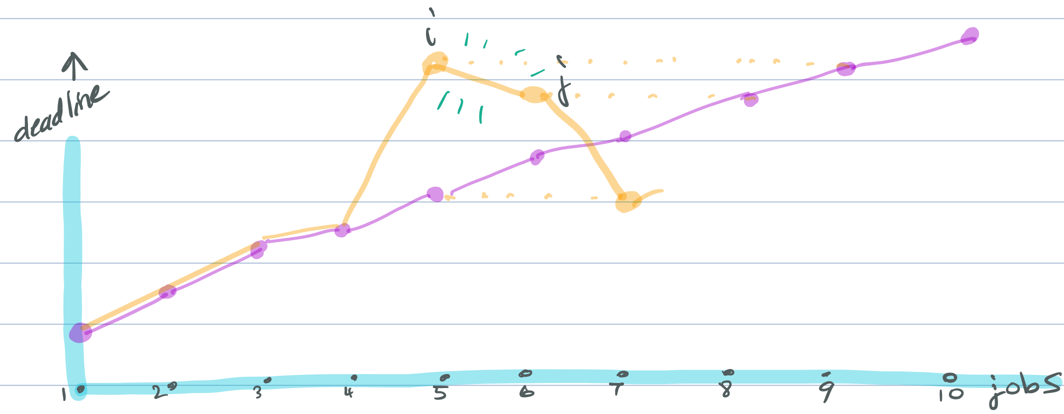
Def$^{\underline{n}}$: Let $I_1, I_2, \ldots, I_n$ be the intervals in sorted order of deadlines, thus defining f_i and l_i for i, $1 \leq i \leq n$
This will be our canonical schedule. I

Def$^{\underline{n}}$: Let $J_1, J_2, \ldots, J_n$ be some other order, defining f_i' and l_i' - this is the schedule J

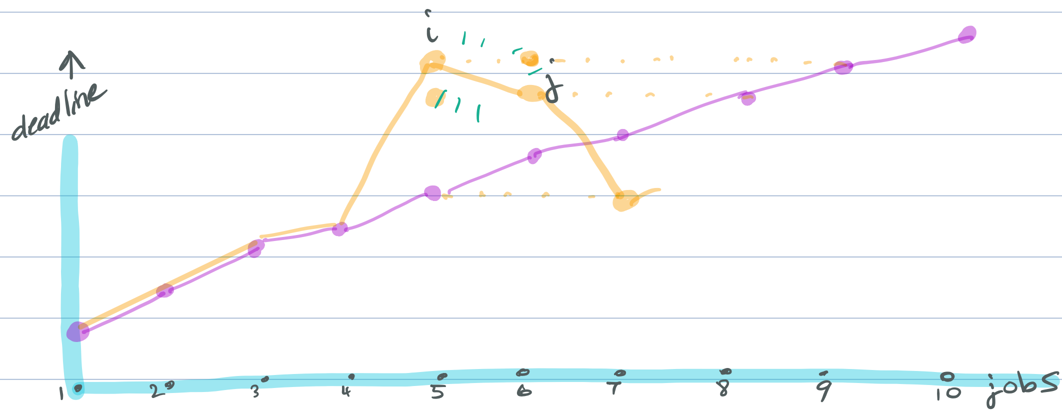
Def$^{\underline{n}}$: An inversion is a pair of jobs in a schedule where the earlier job has the later deadline.

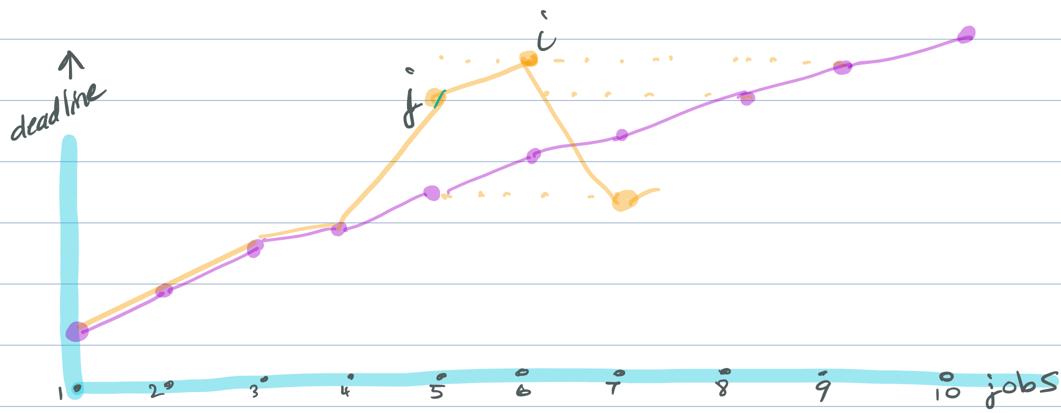
Let us assume for now that $\nexists$ duplicate deadlines.

Claim: $\forall$ schedule $J \neq I$, $\exists$ an adjacent inversion that can be swapped, yielding J^*, where J^* has max latency $\leq$ that of J.



- if J has an inversion, it has an adjacent one.
- let the earliest adjacent inversion be at job i, followed by job j.





difference in latency:

i's latency now is <u>less</u> than j's latency before.

j's latency now is <u>less</u> that j's latency before.

j's latency before was the one that might be max.