Oct 12

Scheduling to minimize maximum lateness

Input: n jobs, i^{th} job (d_i, t_i) (deadline, duration), $1 \le i \le n$

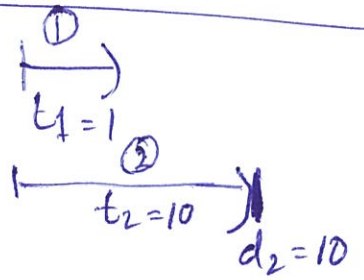
Output: $\forall\ 1 \le i \le n$ compute $s(i)$
$\Rightarrow$ i is scheduled from $[s(i), f(i))$ $\{s(i), \ldots, f(i)-1\}$
where $f(i) = s(i) + t_i$
$\rightarrow$ None of the $[s(i), f(i))$ are in conflict.

Goal: Minimize max lateness

lateness of job i, $l_i = \max(0, f(i) - d_i)$

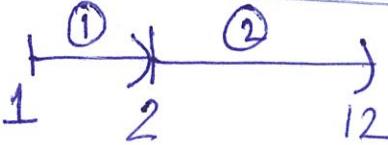
RELU$(x) = \max(0, x)$

Want valid schedule S that min $L(S) = \max l_i$



$d_1 = 100$

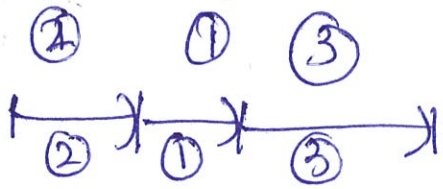
$L = \max(0, 1) = 1$



$l_1 = \max(0, 2 - 100 - 1) = \max(0, -99) = 0$

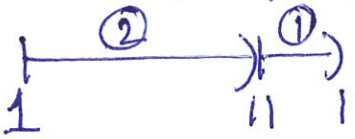
$l_2 = \max(0, 12 - 10 - 1) = 1$

Greedy algo template: Order the jobs & then schedule them in order



→ shortest job first

Schedule 2:



$l_1 = \max(0, 12 - 100 - 1) = 0$

$l_2 = \max(0, 11 - 10 - 1) = 0$

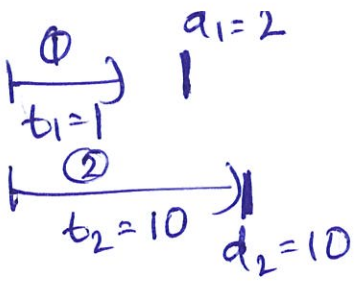
$\boxed{L = 0}$

Suggestions: (i) least slack ($d_i - t_i$) first (at the start smallest $t_i - d_i$ ≡ largest $d_i - t_i$)

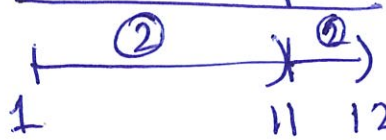
(ii) min lateness first

(iii) Earliest deadline

Ex 2:



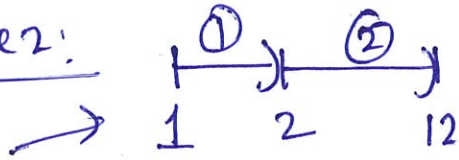
Least slack first:



$l_1 = \max(0, 12 - 2 - 1)$
$= 9$
$l_2 = \max(0, 11 - 10 - 1)$
$= 0$

$L = 9$

Schedule 2:



$l_1 = \max(0, 2 - 1 - 2) = 0$
$l_2 = \max(0, 12 - 10 - 1) = 1$

$L = 1$

earliest deadline first
(largest slack first) ← rule out offline

Earliest deadline first