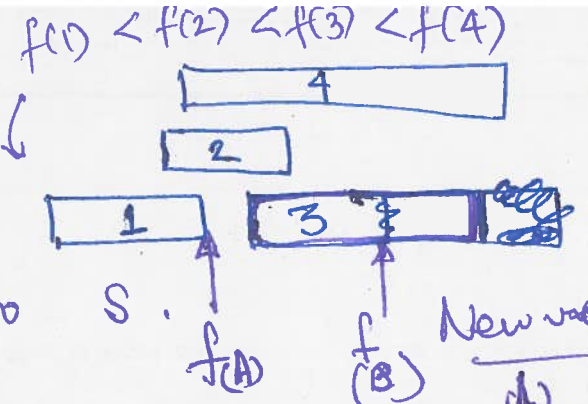
Oct 11



old version:
→ delete 2 & 4
→ Add 3 to S

New version:
(A) f = f(1)
(B) skip over 2 as s(2) < f
(C) f = f(3)
(D) skip over 4 as s(4) < f.

Scheduling to minimize max lateness

Input: n jobs. i^{th} job: $i \in [n]$ (d_i, t_i)
d_i → deadline, t_i ← duration

Output: $\forall\, i \in [n]$, compute $s(i)$
$\Rightarrow$ i is scheduled from $[s(i), f(i))$
$= s(i) + t_i$ } call this f_i

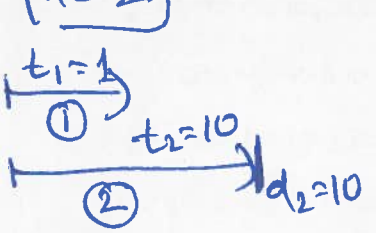
(NOTE: No overlaps are allowed)

Goal: minimize max lateness

lateness of job i : $\max(0, f(i) - 1 - d_i)$

For a schedule S, max lateness $L(S) = \max_i l_i$

Ex 1: $n = 2$



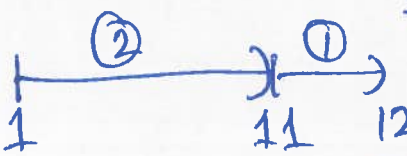
Schedule (1) {smallest duration first}

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

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

L = 1

Schedule 2:

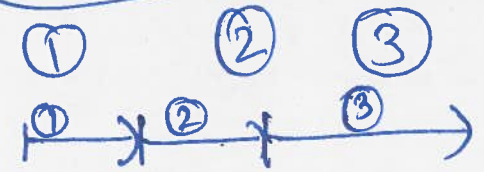


$l_1 = \max(0, 12-1-100) = 0$

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

L = 0

Greedy Algo template: ① Order the jobs → schedule them one after the other



[Q] What kind of orders to consider?

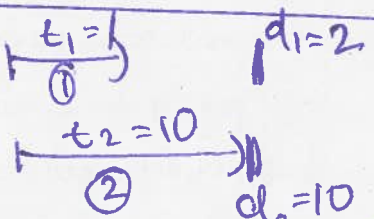
A1: smallest duration first ✗

A2: Largest duration first ✗

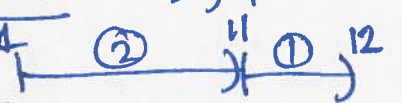
A3: Earliest deadline first ✓

A4: ~~Smallest~~ slack $(d_i - t_i)$ ✗

Ex2: $n=2$



Schedule: 2, 1

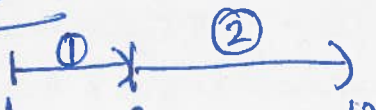


$\ell_1 = \max(0, 12-1-2) = 9$

$\ell_2 = \max(0, 11-1-10) = 0$

$L = 9$

Schedule 2:



$\ell_1 = \max(0, 2-1-2) = 0$

$\ell_2 = \max(0, 12-1-10) = 1$

$L = 1$

Earliest Deadline first algo

0. Sort jobs by deadline ($\Rightarrow d_1 \le d_2 \le \cdots \le d_n$)
1. ~~S~~ $f = 1$
2. for $i = 1 \ldots n$
   - (2.1) $s(i) = f$
   - (2.2) $f(i) = s(i) + t_i$
   - (2.3) $f = f(i)$
3. Return the schedule: $i \mapsto [s(i), f(i))$.