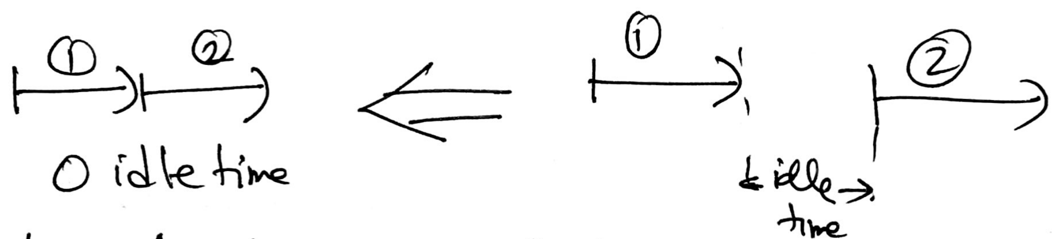
$\boxed{\text{Oct 14}}$ Let A be the greedy schedule
Let $\mathcal{O}$ be an optimal schedule
[$L(S)$ = max lateness of schedule S]

THM 1: $L(A) = L(\mathcal{O})$

Def: Idle time of a schedule is max gap between 2 consecutively scheduled jobs.



Obs 1: A has 0 idle time

Obs 2: Can assume that $\mathcal{O}$ has 0 idle time
("Squish" the gaps $\Rightarrow$ finish times do not increase
$\Rightarrow l_i$ do not increase $\forall i \Rightarrow L(\mathcal{O})$ doesn't increase)

Def: Inversion Given a schedule S & two jobs i & j; (i, j) is an inversion if $d_i > d_j$ but i is scheduled before j [$S(i) < S(j)$]

Obs 3: A has 0 inversions (by algo definition)

Lemma 1: If S_1 & S_2 have both 0 idle time and 0 # of inversions $\Rightarrow$ $L(S_1) = L(S_2)$

Lemma 2: A has 0 idle time & 0 # inversions (Obs 1+3)

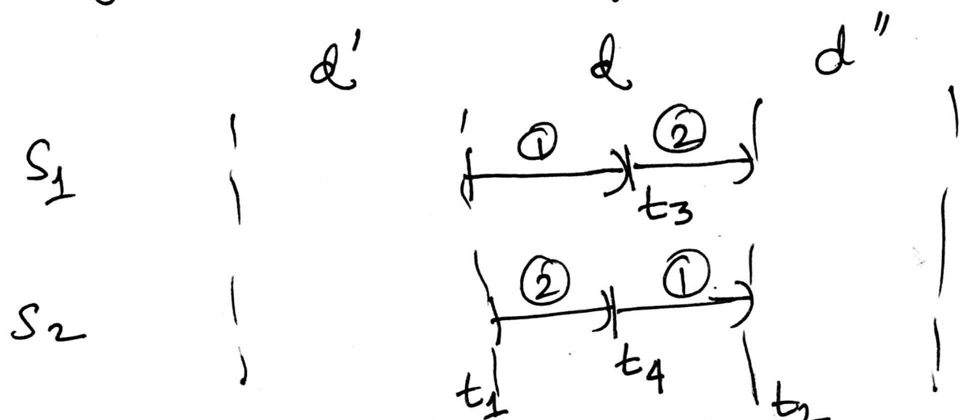
Lemma 3: $\exists$ an $\mathcal{O}$ with 0 idle time & 0 # inversions

Lemmas 1+2+3 $\Rightarrow$ THM 1 $\blacksquare$

<u>Pf (idea) of Lemma 1</u>: S_1 & S_2 both have 0 idle time & 0 # inversions.

<u>Claim</u>: For any schedule with 0 idle time & 0 # inversions, for any deadline d, $\forall i$ s.t. $d_i = d$ are scheduled right next to each other in exactly the same time range

$d_1 = d_2 = d$



$d' < d < d''$

Assume Claim is true

$\Rightarrow$ In S_1: $\left.\begin{aligned} l_1 &= \max(0, t_3 - d - 1) \\ l_2 &= \max(0, t_2 - d - 1) \end{aligned}\right\}$ $\max(l_1, l_2) = l_2 = \max(0, t_2 - d - 1)$

$\Rightarrow$ In S_2: $\left.\begin{aligned} l_1' &= \max(0, t_2 - d - 1) \\ l_2' &= \max(0, t_4 - d - 1) \end{aligned}\right\}$ $\max(l_1', l_2') = l_1' = \max(0, t_2 - d - 1)$

$\Rightarrow$ max lateness among all jobs with same deadline d is <u>same</u> in S_1 & S_2

$\Rightarrow$ (since choice of d was arbitrary) $L(S_1) = L(S_2)$ ∎

Pf (idea) of Claim:

0 # inversions $\Rightarrow$ all jobs with deadline d are scheduled one after the other

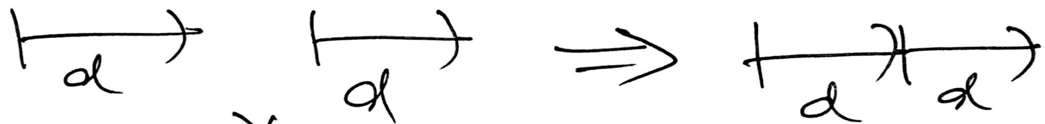
If not:

$d' \neq d$

If $d > d'$ inversion ✗

If $d' > d$ $\Rightarrow$ an inversion ✗

0 idle time $\Rightarrow$ no gap b/w two consecutive jobs.



✗

$f(n) = n^n$, $g(n) = 10^{100n}$ @ $x = 2^{\log_2 x}$

$\|$

$(2^{\log n})^n = 2^{n \log n}$ vs $2^{\log 10 \cdot 100 \cdot n}$

for large n, $\log n > 100 \cdot \log 10$