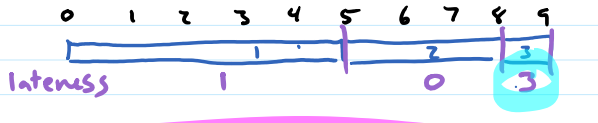
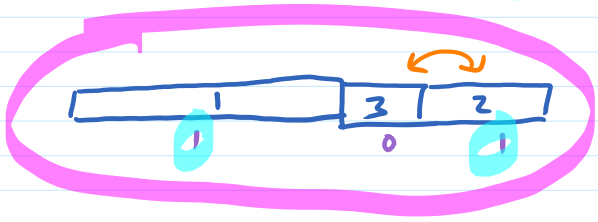
Minimizing Lateness

Given requests with lengths $t_1, \ldots, t_n$, deadlines $d_1, \ldots, d_n$,
find schedule that minimizes maximum lateness.

Ex:

i	t_i	d_i
1	5	9
2	3	8
3	1	6

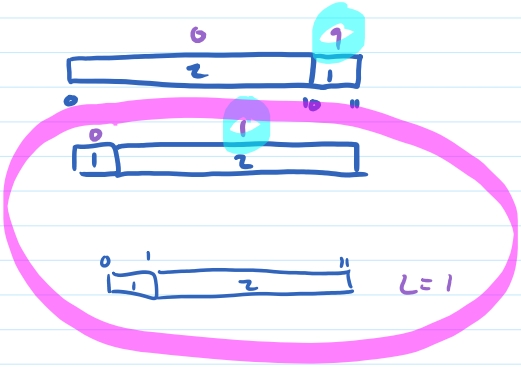




shortest first?

most pressing first?
break ties by shortest

i	t_i	d_i
1	1	2
2	10	10



earliest deadline first?

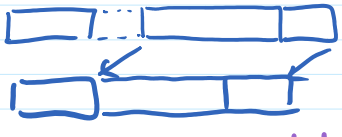
1) There is an optimal schedule with no idle time

$\max(l_1, \ldots, l_n) \geq \max(\bar{l}_1, \ldots, \bar{l}_n)$

$\bar{l}_i \leq l_i$

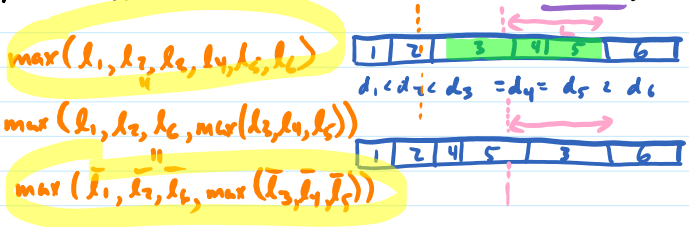
$O =$ optimal schedule w/ idle time

$\bar{O} =$ latenesses stay same or go down



2) All schedules with no idle time and no inversions have same max lateness

task i started before j but with $d_i > d_j$



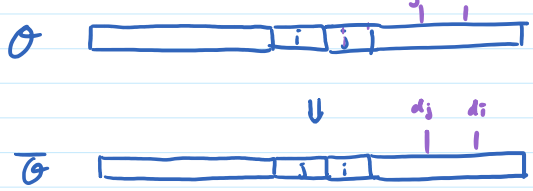
3) There is a optimal schedule with no inversions, no idle time.

(schedule back-to-back in order of ↑ deadline)

So: greedy has no inversions, no idle time
there is optimal O $\forall$ no inversions, no idle time 3
greedy has same max lateness as O 2
greedy is also optimal

Can find opt sched O with no idle time (1)

If O has no inversions then ü
Else find inversion between adj in schedule

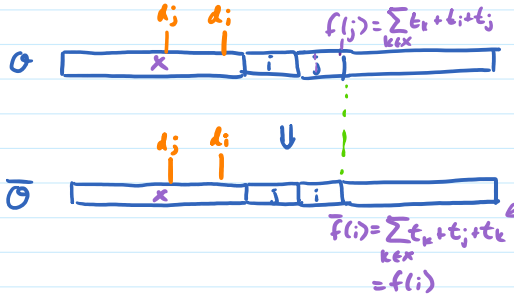


finish time of j

if $d_j \geq f(j)$ $\ell_i = 0 = \ell_j$

$\bar{\ell}_i = 0 = \bar{\ell}_j$

$\max(\ell_1, \dots, \ell_n) = \max(\max(\ell_i, \ell_j), \max(\text{other lateness}))$
$= \max(\max(\bar{\ell}_i, \ell_j), \max(\text{other lateness}))$



$\ell_i = f(i) - d_i$
$\ell_j = f(j) - d_j$

$\bar{\ell}_i = \bar{f}(i) - d_i$
$= f(j) - d_i$
$< f(j) - d_j = \ell_j$

$\bar{\ell}_i < \ell_j$

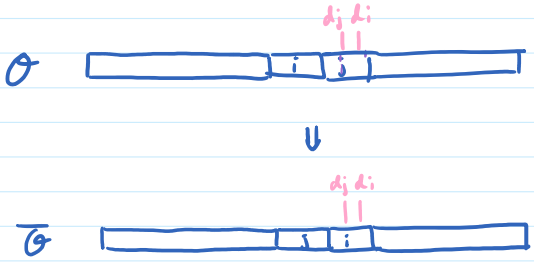
We $d_i > d_j$

$\bar{\ell}_i = \ell_j - t_i < \ell_j$

$\max(\bar{\ell}_i, \bar{\ell}_j) \leq \max(\ell_j, \ell_j - t_i) = \ell_j \leq \max(\ell_i, \ell_j)$

max of smaller terms is smaller

def. max



$\ell_i = 0$
$\ell_j = f(j) - d_j$
$\max(\ell_i, \ell_j) = f(j) - d_j$

$\bar{\ell}_j = 0$
$\bar{\ell}_i = \bar{f}(i) - d_i$
$= f(j) - d_i$
$< f(j) - d_j = \ell_j$

$\max(\bar{\ell}_i, \bar{\ell}_j) = \bar{\ell}_i < \ell_j = \max(\ell_i, \ell_j)$

No matter where d_i, d_j are, we have

"moves" ↓ so if $y > x$

No matter where d_i, d_j are, we have

$$\ell_j = f(j) \dot{-} d_j$$
$$\ell_i = f(i) \dot{-} d_i \leq f(j) \dot{-} d_i \leq f(j) \dot{-} d_j = \ell_j$$

($f(i) < f(j)$, $d_j \leq d_i$)

"monus"

$$x \dot{-} y = \begin{cases} 0 & \text{if } y \geq x \\ x - y & \text{otherwise} \end{cases}$$

to avoid monus, see cases on next page

$$a > b \rightarrow \begin{cases} x - a < x - b \\ x \dot{-} a = x \dot{-} b \end{cases}$$

so $\max(\ell_i, \ell_j) = \ell_j$ (j was the later of the two in original optimal schedule $\mathcal{O}$)

$\bar{f}(j) < f(j)$ & $\bar{f}(j) = f(j) - t_i$

$$\bar{\ell}_j = \bar{f}(j) \dot{-} d_j \leq f(j) \dot{-} d_j = \ell_j$$

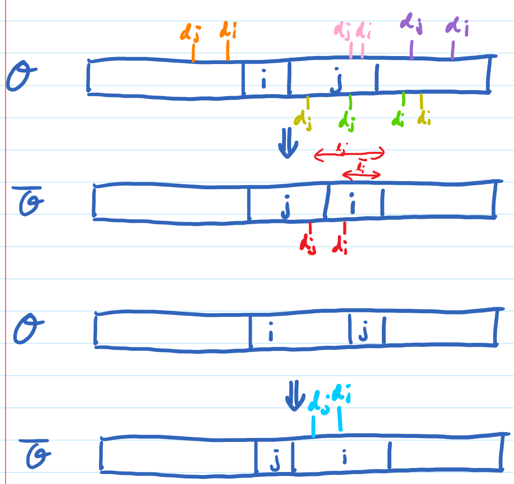
$$\bar{\ell}_i = \bar{f}(i) \dot{-} d_i = f(j) \dot{-} d_i \leq f(j) \dot{-} d_j = \ell_j$$

$$\max(\bar{\ell}_i, \bar{\ell}_j) \leq \max(\ell_j, \ell_j) = \ell_j = \max(\ell_i, \ell_j)$$

$\bar{\ell}_i \leq \ell_j$, $\bar{\ell}_j \leq \ell_j$

max of smaller things is smaller

All possible cases



in O: i	in O: j	in Ō: i	in Ō: j	
N	N	N	N	$\max(\bar{l}_i, \bar{l}_j) = \max(0,0) = \max(l_i, l_j)$
~~N~~	~~N~~	~~Y~~	~~N~~	j not late → i can't become late
N	Y	N	N	$\max(\bar{l}_i, \bar{l}_j) = 0 < l_j = \max(0, l_j) = \max(l_i, l_j)$
N	Y	N	Y	max lateness = j's lateness, which ↓
N	Y	Y	N	DONE on other slide
N	Y	Y	Y	same as ↓
Y	Y	Y	N	
Y	Y	Y	Y	DONE on other slide

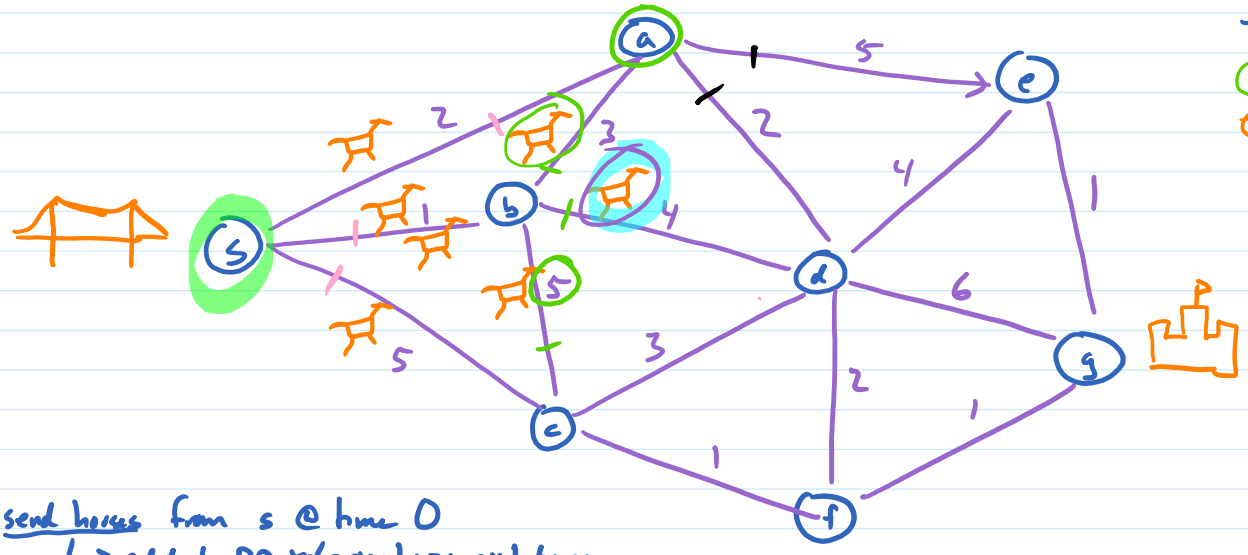
$$\max(\bar{l}_i, \bar{l}_j) = \max(\bar{l}_i, l_j - t_i) \le \max(l_j, l_j - t_i) = \max(l_j) = \max(l_i, l_j)$$

↑ in YY→YY case this is ≤

$$\max(\bar{l}_i, \bar{l}_j) = \bar{l}_i = \bar{f}(i) - d_i = f(j) - d_i < f(j) - d_j = l_j \le \max(l_i, l_j)$$

↑ $\bar{l}_j = 0$ ↑ def ↑ $\bar{f}(i) = f(j)$ ↑ $d_i > d_j$ ↑ def ↑ max

Given weighted graph with no negative-weight edges and start vertex s, find path of minimum total weight from s to all other vertices



Priority Queue

a	s	2
b	s	1
c	s	5
d	a	5
e	a	7

4

send horses from s @ time 0
↳ add to PQ w/ priority = arrival time
also record start/destination

while PQ not empty (lowest priority (time))
remove next event (horse arrival) from PQ
time gives total weight of shortest path
node horse came from gives next-to-last vertex
send new horses (add new nodes to PQ)
↳ don't send to nodes already taken off PQ
don't send to a node if a prev horse is getting there sooner
cancel prev horse if new horse gets there sooner
update priority (time) and source for that node