Greedy
Algorithm

Weighted Interval Scheduling

have a value v_i for each

$\max_{A} \sum_{i \in A} v_i$ s.t. A non-overlapping. : Thm. Cannot solve problem w/ a greedy algo.

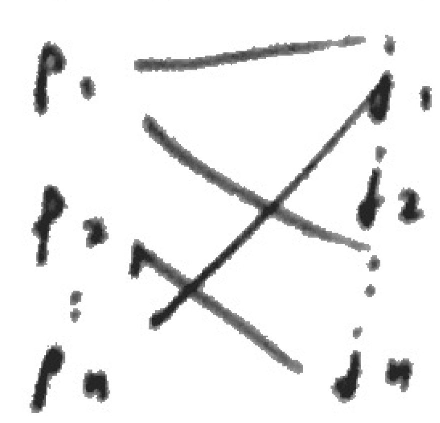
but by dynamic programming

Matching : $p_1 \cdots p_n$

$j_1 \cdots j_n$

some people can do some jobs.

want to match as many people as possible :



$m = \#$ of edges. $O(m \cdot n^{1/2})$, $O(m^{10/7})$

edge (p_a, j_b) if p_a can do job j_b

solve by flow algorithm :

CPSC 365 Lecture 4 1/22

Two more scheduling problems

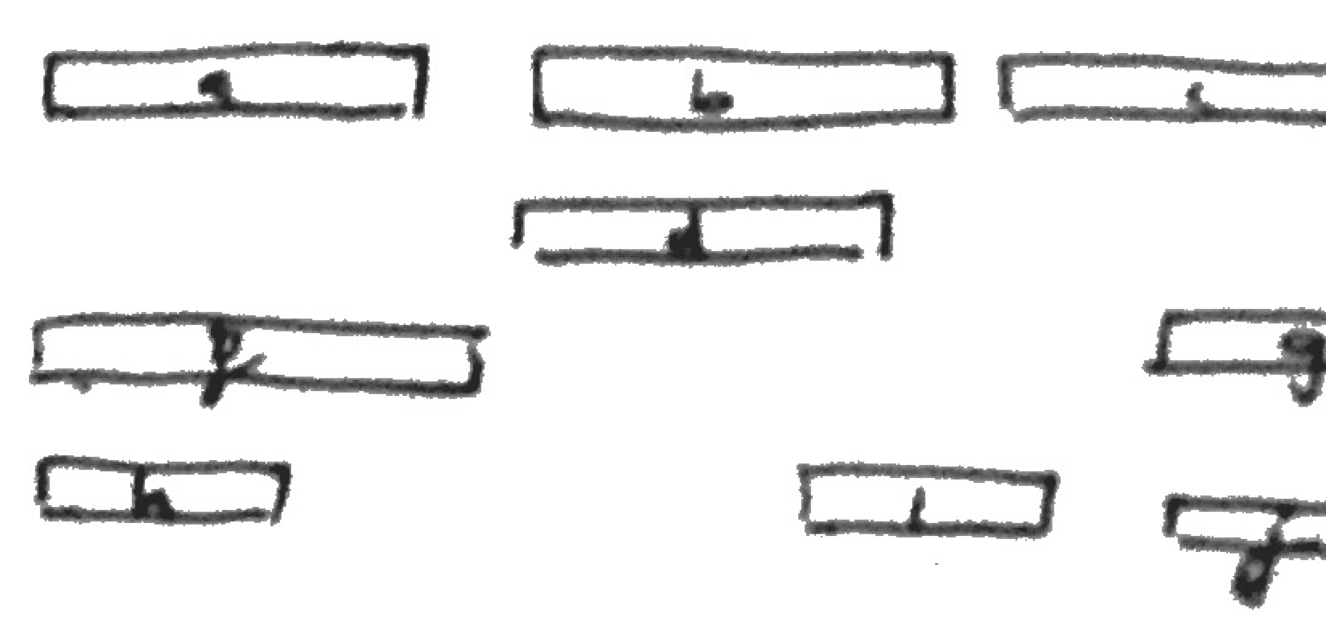
1. Class scheduling

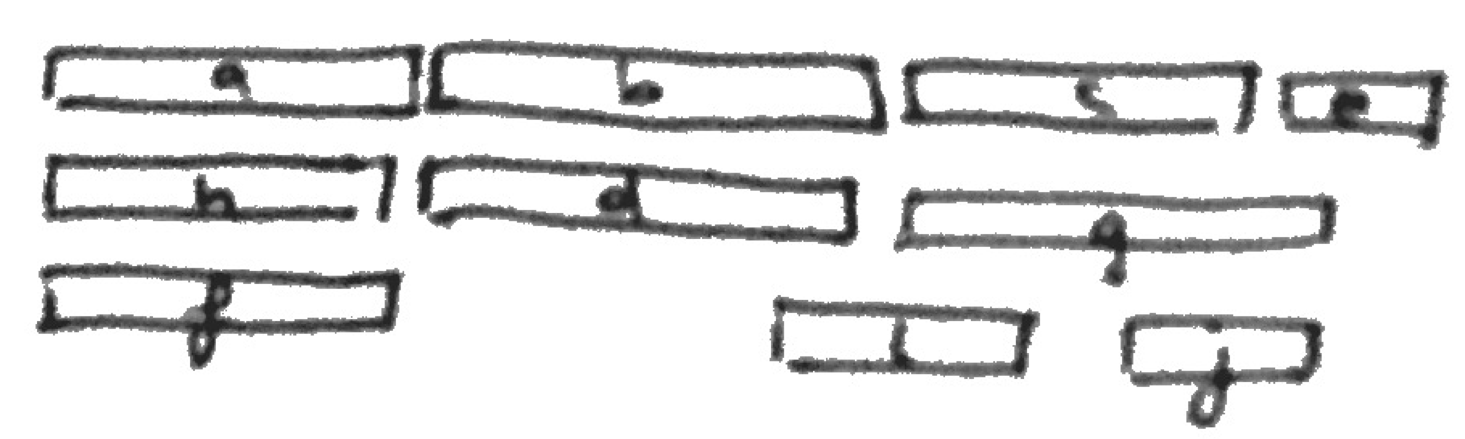
input: time intervals.

interval i from $s(i)$ to $f(i)$

problem

1. Schedule in the available rooms.
2. Schedule using as few rooms as possible





for any time t, define

$ply(t) = |\{i : s(i) < t < f(i)\}|$

rooms need $\geq ply(t)$

$max\text{-}ply = \max_t ply(t)$

rooms need

$\geq max\text{-}ply$

rooms

Gre

Greedy Algorithm: → sort so that $s(1) \le s(2) \dots \le s(n)$

<u>init</u> $k=0$ (# rooms used)

for $i=1$ to n, $r(i)=\emptyset$ (room assigned to class i)

for $i=1$ to n

if $\exists$ a room $l \in \{1 \dots k\}$ so that i can fit class i into room l

($\exists l$ s.t. $\forall j$ s.t. $r(j)=l$, classes j and i do not overlap)

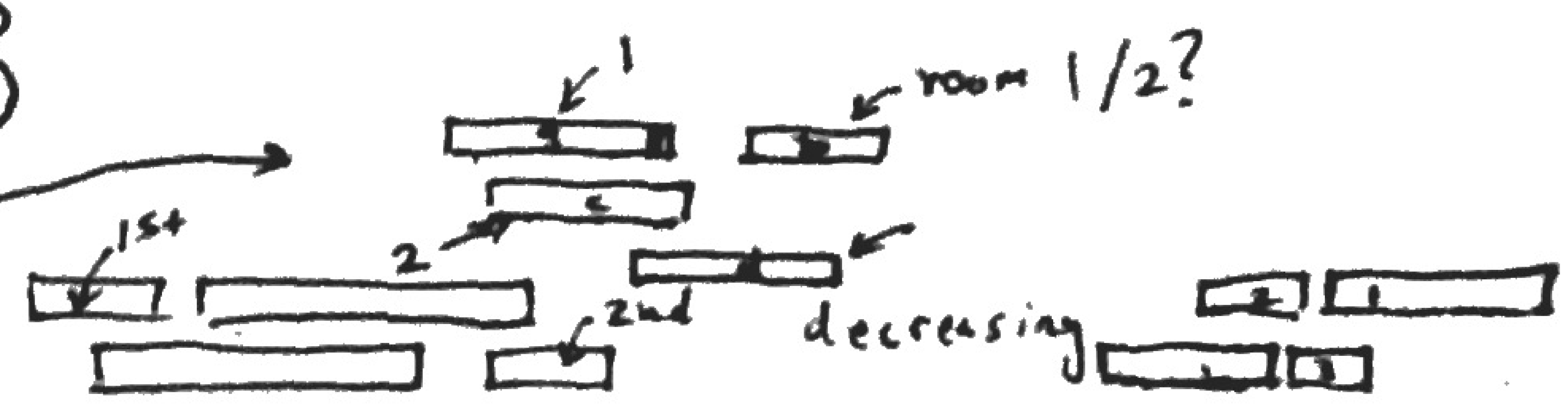
let l be such a room, set $r(i)=l$

else, $k=k+1$,

$r(i)=k$

<u>Orders</u>

- increasing start time ✓ まる ○
- increasing finish time →
- increasing class time →
- decreasing overlaps.



<u>Theorem</u> if $s(1) \le s(2) \le \dots \le s(n)$

then alg finishes with $k \le$ max-ply.

$A_i = \{j : j < i$ and class j overlaps class $i\}$

<u>Lem</u> for all i, $|A_i| \le$ max-ply -1

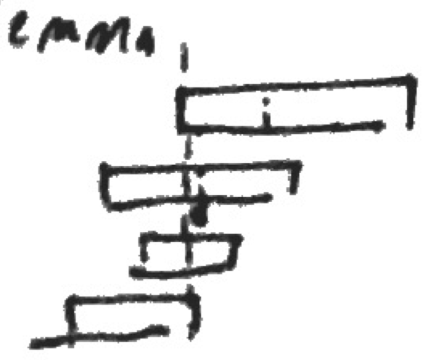
<u>proof of theorem</u>

if $k=$ max-ply, I claim k will not increase. When looking at class i, only conflicts with classes in A_i.

if A_i is smaller then k, then there is a free room to place class i. ($|A_i| < K$, then is an $l \in \{1 \dots k\}$, not occupied)

proof of lemma,

$j \in A_i$



$t = \frac{1}{2}(s(i) + \bigvee_{j \in A_i} f(j))$

$t = s(i) + \varepsilon$

$\Rightarrow$ ply$(t) \ge 1 + |A_i|$

$|A_i| \le$ ply$(t) - 1$

$\le$ max-ply -1

Lec. 4 1/22

2. Minimizing lateness

have n jobs — choose $s(i)$ = start of job i.

deadline $d(i)$ — finishes $t(i)$ time later $f(i) = s(i) + t(i)$

time takes $t(i)$ — not allowed to overlap.

lateness $\ell(i) = \max(f(i) - d(i), 0)$ — max-lateness $\stackrel{\text{def}}{=} \max_i(\ell(i))$

Goal: minimize maximum lateness. $\Rightarrow$ $\vee(\wedge \ell(i))$ = try to control the worst impression we

t	d	s	f	l	s	f	l
3	2	0	3	1	0	3	1
4	8	3	7	0	5	9	1
2	7.	7	9	2	3	5	0

solution

do jobs in the order of deadline.

Sol | do job in order of deadline, sort so that $d(1) \le d(2) \le \dots \le d(n)$

Theorem max lateness$(1, 2 \dots n) \le$ max-lateness$(\pi(1), \dots, \pi(n))$

do $\pi(1)$ first, then $\pi(2)$

proof: let π be an order other than $1, 2 \dots n$

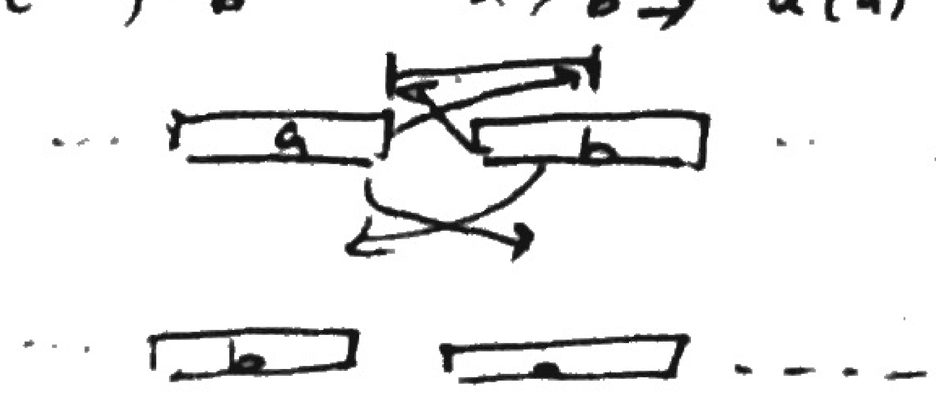
then is an i so that $\pi(i) > \pi(i+1)$

swap the order of those two $\Rightarrow$ will not increase max-lateness.

$\pi(i) = a$ $\quad$ $\pi(i+1) = b$ $\quad$ $a > b \Rightarrow d(a) \ge d(b)$

π

$\hat{\pi}$



deadline of a is after deadline of b.

$\ell\pi(b) \ge \ell\pi(a)$ $\qquad$ $\ell\hat{\pi}(a) - \ell(a)$

$\ell\hat{\pi}(b) \le \ell\pi(b)$ $\qquad$ $= f\pi(b) - d(a)$

$\ell\hat{\pi}(a) \le \ell\pi(b)$ $\qquad$ $\cdot f^{\pi}(b) - \ell(b)$

a & b are only two that move

so max-lateness$(\hat{\pi}) \le$ max-lateness(π)

if $\hat{\hat{\pi}} \le$ max-$\ell(\hat{\pi})$

max-$\ell(\hat{\hat{\pi}}) \le$ max $\ell(\pi)$ $\qquad$ $|A_i| \le$ ply$(P) - 1$

$\le$ max-ply $- 1$