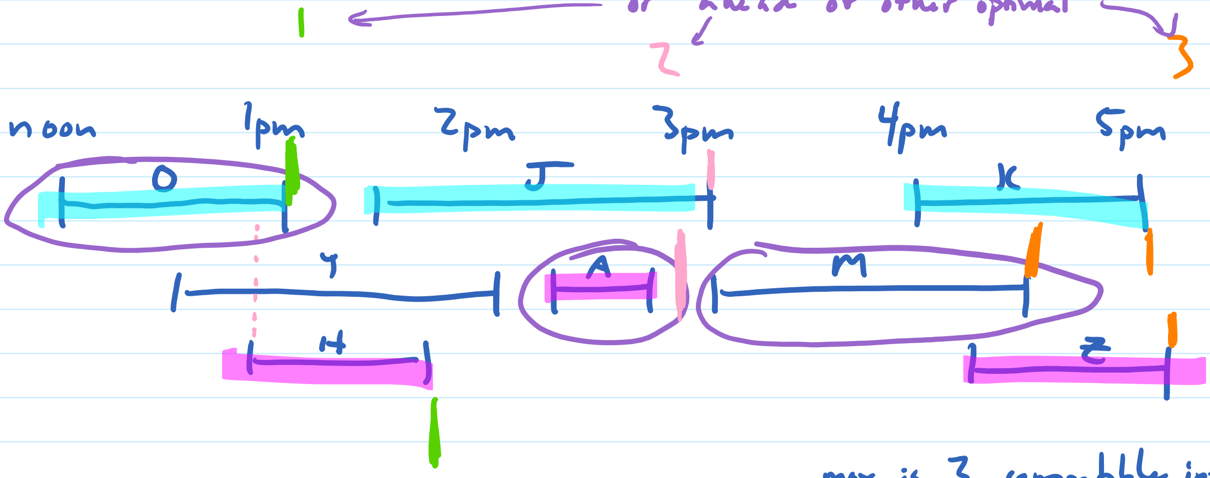
Continuing Education Credits

greedy always on pace with or ahead of other optimal



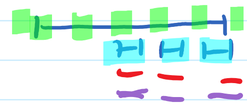
max is 3 compatible intervals

Interval Scheduling

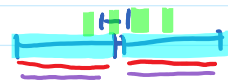
Given intervals labelled $1, \ldots, n$ w/ interval i's start, finish = $s(i)$, $f(i)$,
find largest set of pairwise compatible intervals

interval i, j $(i \neq j)$
compatible if $s(i) \le f(i) \le s(j) \le f(j)$
or vice versa

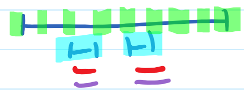
Greedy: choose 1st start? No:



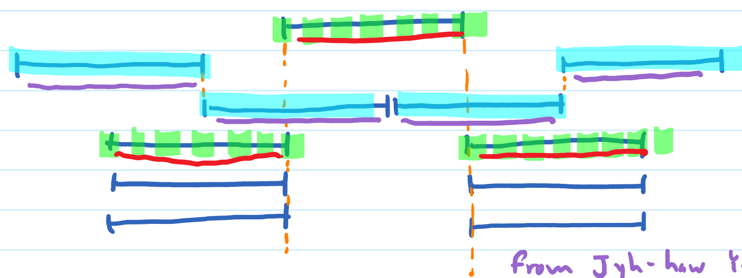
choose shortest? No



choose last finish? No



fewest overlaps? works for prev examples, but no:



from Jyh-haw Yeh,
Boise State

choose 1st to finish...

Interval Scheduling

Given intervals labelled $1, \dots, n$ w/ interval i's start, finish $= s[i], f[i]$,
find largest set of pairwise compatible intervals

↳ interval i, j $(i \neq j)$
compatible if $s(i) \le f(i) \le s(j) \le f(j)$ or vice versa

Greedy algorithm:

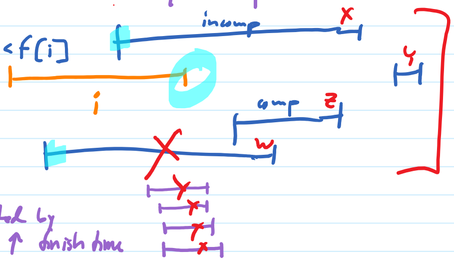
sort R in order of increasing finish $O(n \log n)$

$R \leftarrow \{1, \dots, n\}$ — intervals compatible with A
$A \leftarrow \emptyset$ — set of chosen intervals — next ← 1
while $R \neq \emptyset$ — something left to pick — next ≤ n
  choose $i \in R$ to minimize $f[i]$ — i is 1st to finish among compatible with A — i ← next
  $A \leftarrow A \cup \{i\}$
  remove from R all incompatible with i — update R for newly incompatible
return A

$O(n)$ [while next ≤ n and s[R[next]] < f[i]
  next ← next + 1]

$O(n)$ total

$O(n \log n)$ overall



Proof of optimality: let $\mathcal{O} = j_1, \dots, j_m$ — sorted by ↑ finish time

kth in A — kth in optimal soln $\mathcal{O}$

INVARIANT: after iteration k, $f[a_k] \le f[j_k]$ — at or ahead of optimal $\mathcal{O}$
A all pairwise compatible and sorted
R is intervals compatible with A
$|A| = k$

Init: trivial

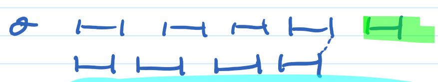
Maintenance: Suppose after k iterations, INV is true and $R \neq \emptyset$
[need $f[a_{k+1}] \le f[j_{k+1}]$]

$f[j_{k+1}] > f[j_k]$ — $\mathcal{O}$ ordered by ↑ finish
$s[j_{k+1}] \ge f[j_k]$ — otherwise j_{k+1} overlaps j_k
$f[a_k] \le f[j_k]$ — INV

$$\underline{s[j_{k+1}] \ge \underbrace{f[j_k] \ge}_{\text{INV}} \overbrace{f[a_k] > s[a_k] \ge f[a_{k-1}] > s[a_{k-1}] \ge \dots \ge s[a_1]}^{\text{A is sorted}}}$$

j_{k+1} is compatible with A

$f[a_{k+1}] \le f[j_{k+1}]$ — $\overbrace{a_{k+1} \text{ has min finish of all in R}}^{\text{code}} = \underbrace{\text{all compatible with A}}_{\text{INV}}$,



Termination: loop stops after m iterations

can stop after $k < m$ iterations? NO — j_{k+1} is compatible with $a_1, \dots, a_k$ b/c $s[j_{k+1}] \ge f[j_k] \ge f[a_k]$
so $j_{k+1} \in R$
$R \neq \emptyset$
so loop didn't stop

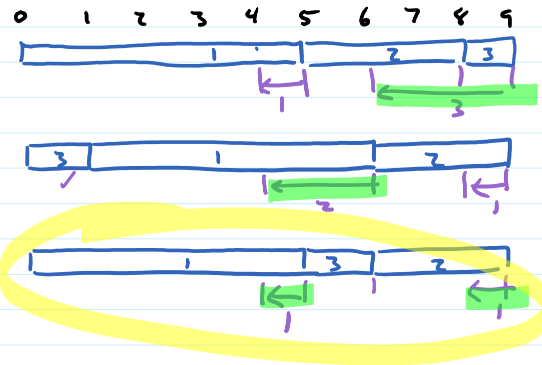
also can't stop after $k > m$ iterations (would have bigger soln than opt $\mathcal{O}$)
∴ must stop after exactly m iterations

Post Condition: at termination, $|A|$ is a set of pairwise compatible intervals of size m, same as $\mathcal{O}$, so A is optimal too

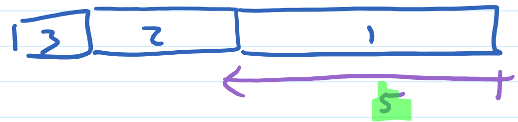
Given requests with lengths $t_1, \ldots, t_n$, deadlines $d_1, \ldots, d_n$, find schedule that minimizes maximum lateness.

Ex:

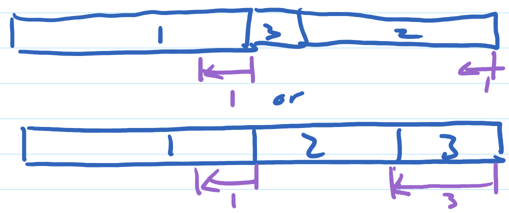
i	t_i	d_i
1	5	4
2	3	8
3	1	6



shortest first?



most pressing first?
latest if done next
($\max t_{curr} + t_i - d_i$)



so break ties by shortest?