Job scheduling

Interval scheduling

Greedy strategy, sort by earliest finish time

Deadline scheduling

Each job i has deadline $d(i)$, time to process $t(i)$

Schedule all jobs

A job is late if it finishes after $d(i)$

lateness is finish time $- d(i)$

Minimize maximum lateness

Different from what we mentioned earlier

Greedy strategy

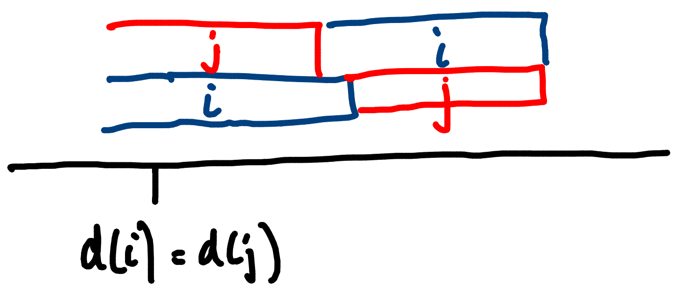
Process in ascending order of $d(i)$

Prove correctness?

Consider some optimal schedule O, our greedy schedule G

- Case 1, O is in ascending order of $d(i)$ but $O \neq G$

Must have some $d(i) = d(j)$, $i \neq j$



Max lateness not affected (sum of lateness may not be preserved!)

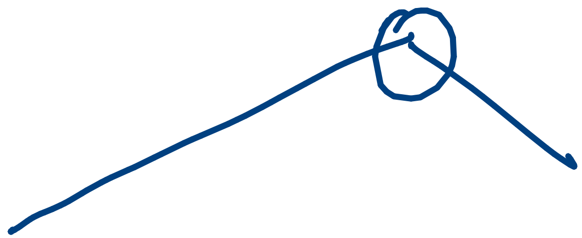
- Case 2: O is not in ascending order of d(i)

<u>Aside:</u> Can assume no gaps in O or G

O must have at least one "inversion"

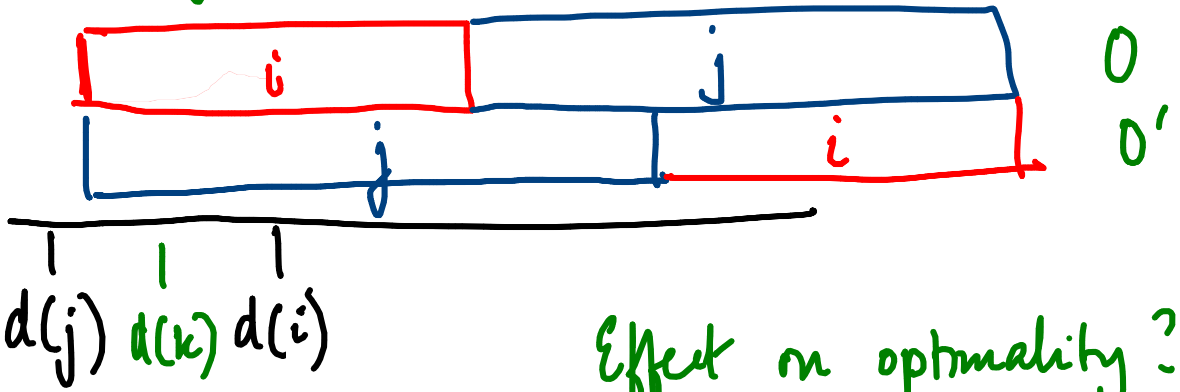
d(i) > d(j), i before j in O

Can we assume i & j are consecutive?



Must be a point where order reverses

Remove adjacent inversion



Effect on optimality?

lateness of j ↓

lateness of i ↑

└ compare to previous lateness of j

$d(i) > d(j)$ so finish time$(j) - d(j)$ in O

$\qquad \qquad > $ finish time$(i) - d(i)$ in O'

At each step

Remove one inversion, preserving optimality

Eventually zero inversions ⇒ same as G

modulo jobs with equal deadlines

"Exchange argument"

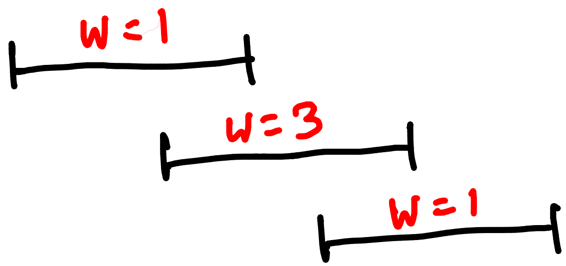
Massage any optimal solution to look like

greedy solution

Weighted interval scheduling

Each job i has start time s(i)
finish time f(i)
value/weight w(i)

Pick a subset to maximize sum of weights



Earliest greedy strategy fails

No obvious greedy strategy

What to do?

Must try all possibilities !!!

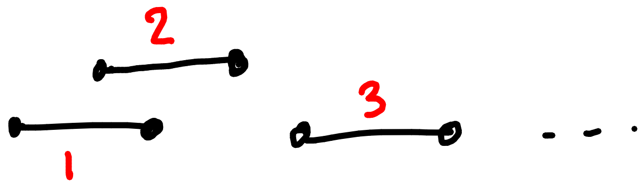
Exploit the problem structure to get an inductive strategy

For example, consider jobs sorted by finish time
$1, 2, \ldots, n$ (after sorting)

Final solution either has job 1 or not

max
Keep Job 1 : $w(1) +$ Solution(X)
$X = \{2, \ldots, n\} \setminus$ jobs in conflict with 1
Drop Job 1 : Solution$(\{2, \ldots, n\})$

Problem

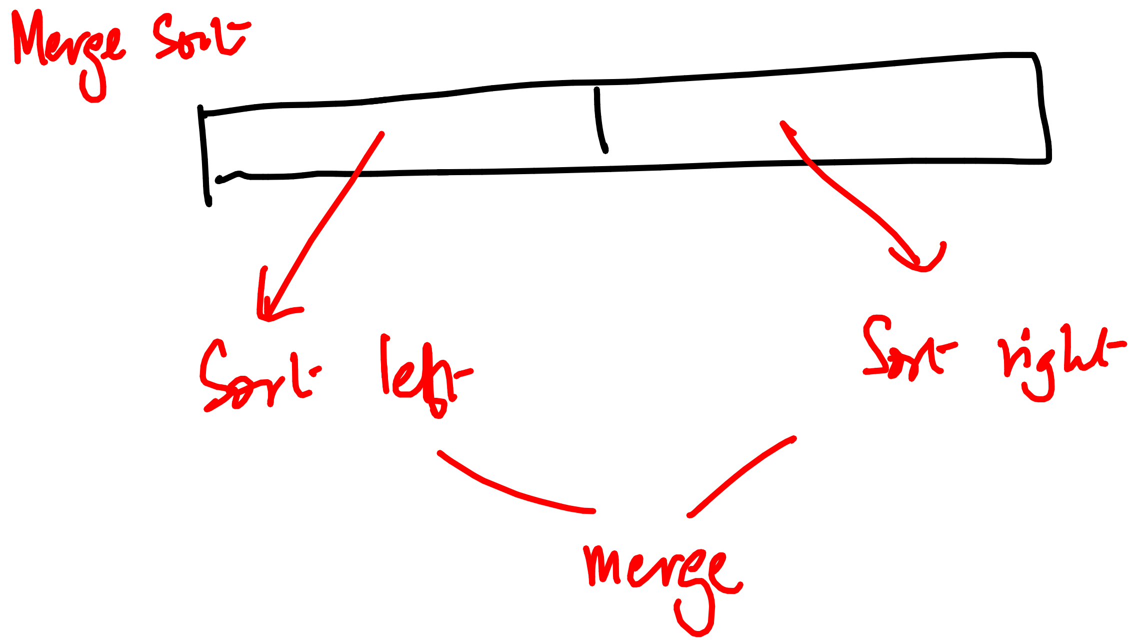


Keep 1 : Solve $(\{3, .., n\})$

Drop 1
- Keep 2 Solve $(\{3, .., n\})$
- Drop 2

Overlapping Subproblems

Contrast to nonoverlapping subproblem



Divide & Conquer