Apr 13 (Weighted Interval Scheduling)

Simplified problem: Only compute the value of an optimal solution

$OPT(J)$ = value of an optimal solution for $[J]$ $\quad 0 \leq J \leq n$

$\{(s_1, f_1, v_1), (s_2, f_2, v_2), \ldots (s_J, f_J, v_J)\}$

Goal: Compute $OPT(n)$

Def: O_J be an optimal solution for $[J]$

$$OPT(J) \underset{\text{by defn of } OPT(J) \text{ \& } O_J}{=} v(O_J)$$

$OPT(0) = 0$

Goal: $OPT(J) = \max\{OPT(J-1),\ v_J + OPT(\underbrace{P(J)}_{\text{TBA}})\}$ $\quad < J$

Case 1: $J \notin O_J$

Claim 1: O_J is also optimal for $[J-1]$

$$\Rightarrow OPT(J) \underset{\text{by defn of } O_J}{=} v(O_J) \underset{\text{by Claim 1}}{=} OPT(J-1)$$

Pf (idea) of Claim 1: Assume O_J is not optimal for $[J-1]$.

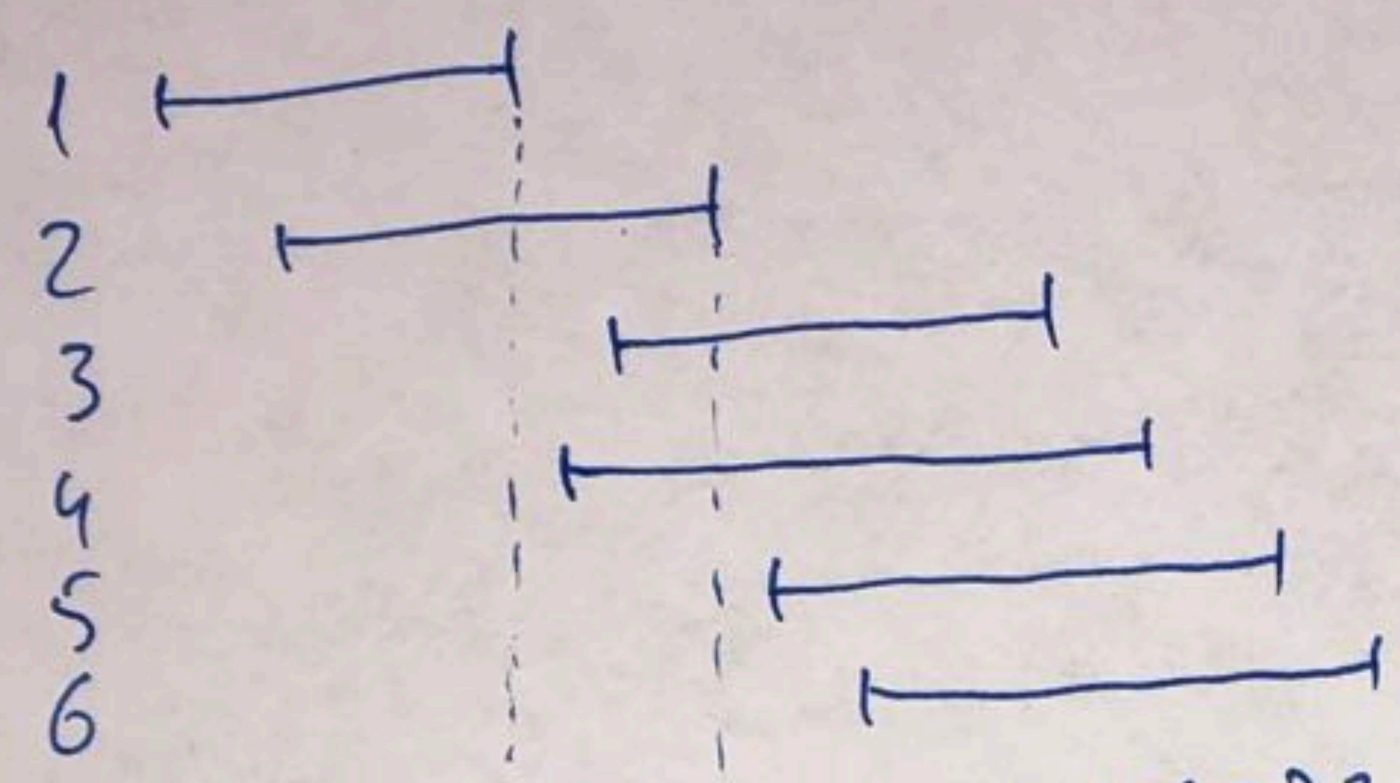
$\exists$ a schedule O' for $[J-1]$ s.t. $v(O') > v(O_J)$

But O' is also valid for $[J] \Rightarrow$ contradicts the assumption that O_J is optimal for $[J]$

Case 2: $j \in \mathcal{O}_j$ | **Def:** $p(j)$ = largest $i < j$ s.t. i & j do not conflict

$= 0$ if no such i $\exists$

Ex.



$p(1) = 0$
$p(2) = 0$
$p(3) = 1$
$p(4) = 1$
$p(5) = 2$
$p(6) = 2$

① $p(j)+1, p(j)+2 \dots j-1$ conflict with j

② $1, 2, \dots, p(j)$ will NOT be in conflict with j

$OPT(6) = \max\{OPT(5), v_6 + OPT(2)\}$

Claim 2: $\mathcal{O}_j \setminus \{j\}$ is an optimal solution for $[p(j)]$

$$\Rightarrow OPT(j) \underset{\text{def of } \mathcal{O}_j}{=} v(\mathcal{O}_j) = v_j + v(\mathcal{O}_j \setminus \{j\})$$

$$\underset{\text{by claim 2}}{=} v_j + OPT(p(j))$$

Pf(idea) of Claim 2: Assume $\mathcal{O}_j \setminus \{j\}$ is NOT optimal for $[p(j)]$

$\Rightarrow \exists\ \mathcal{O}'$ which is a valid schedule for $[p(j)]$ & $v(\mathcal{O}') > v(\mathcal{O}_j \setminus \{j\})$

Note: $\mathcal{O}' \cup \{j\}$ is a valid schedule for $[j]$

but $v(\mathcal{O}' \cup \{j\}) = v(\mathcal{O}') + v_j > v(\mathcal{O}_j \setminus \{j\}) + v_j = v(\mathcal{O}_j)$ for $[j]$

$\Rightarrow$ contradicts optimality of $\mathcal{O}_j$ for $[j]$

Ex. Compute $p(j)$ for $1 \le j \le n$ in $O(n \cdot \log n)$ time.

Bonus. $\Omega(n \cdot \log n)$ comparisons to compute $p(j)$ for all j.