Apr 10 (Weighted Interval Scheduling)

Simplified problem: Instead of outputting an optimal solution $\mathcal{O}$, output $v(\mathcal{O}) = \sum_{i \in \mathcal{O}} v_i$

---

$OPT(J) \overset{def}{=}$ value of an optimal solution for $[J]$ $\begin{cases} (s_1, f_1, v_1) \\ (s_2, f_2, v_2) \\ \vdots \\ (s_J, f_J, v_J) \end{cases}$

$1 \leq J \leq n$

ASSUME: $f_1 \leq f_2 \leq \dots \leq f_n$

Goal: Compute $OPT(n)$

Def: $\mathcal{O}_J$ be an optimal solution for $[J]$

$$v(\mathcal{O}_J) = OPT(J)$$

Case 1: $J \notin \mathcal{O}_J$ Claim 1: $\mathcal{O}_J$ is an optimal solution for $[J-1]$

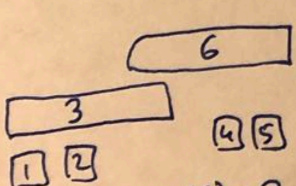
$\Rightarrow OPT(J) = OPT(J-1)$ — (1)

Case 2: $J \in \mathcal{O}_J$ Claim 2: $\mathcal{O}_J \setminus \{J\}$ is an optimal solution for $[p(J)]$

Def: $p(J) =$ ~~smallest~~ largest value $i$ s.t. $i$ & $J$ do not conflict

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

$\Rightarrow OPT(J) = v_J + OPT(p(J))$ — (2)




$p(6) = 2$

Combining (1) & (2) $\Rightarrow OPT(J) = \max\{OPT(J-1), v_J + OPT(p(J))\}$