Nov 8

Weighted interval Scheduling

Input: n intervals, i^{th} interval $= (s_i, f_i, v_i)$

(s_i = start time, f_i = finish time, v_i = value)

Output: Instead of outputting an optimal schedule $\mathcal{O}$ output $v(\mathcal{O}) = \sum_{i \in \mathcal{O}} v_i$

Def: $OPT(j)$ = value of an optimal solution for $[j]$, $1 \le j \le n$

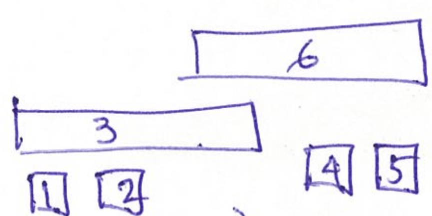
$\{(v_1, f_1, s_1), \ldots, (v_j, f_j, s_j)\}$

$f(1) \le f(2) \le \ldots \le f(n)$

Q: GOAL? A: $OPT(n)$

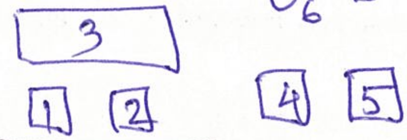
Def: Let $\mathcal{O}_j$ be an optimal solution $[j]$

$v(\mathcal{O}_j) = OPT(j)$



Case 1: $j \notin \mathcal{O}_j$

→ **Ex!** $6 \notin \mathcal{O}_6$



$\mathcal{O}_6$ is an optimal solution for $[5]$

→ $\boxed{OPT(j) = OPT(j-1)}$

Case 2: $j \in \mathcal{O}_j$ → $6 \in \mathcal{O}_6$, $\mathcal{O}_6 \setminus \{6\}$

Claim: $\mathcal{O}_6 \setminus \{6\}$ is an optimal solution for $\{1,2\}$

$OPT(6) = v_6 + OPT(2)$ ← $p(6) = 2$

Def: $p(j)$ = Largest value of $i < j$ s.t. i doesn't conflict with j

$\boxed{OPT(j) = v_j + OPT(p(j))}$

$OPT(j) = \max\{OPT(j-1), v_j + OPT(p(j))\}$