Weighted Interval Scheduling

Input: n intervals; each interval $i = \{s_i, f_i, v_i\}$

s_i = start time

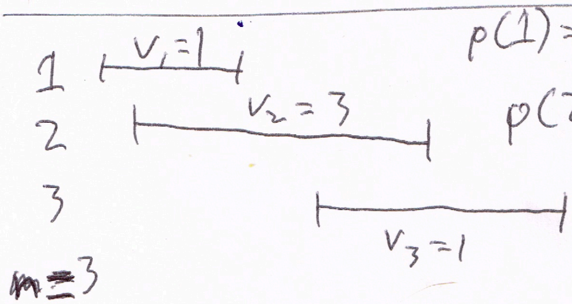
f_i = finish time

v_i = weight

Output: valid schedule $S \subseteq [n] = \{1, \dots, n\}$

valid means $\forall i \neq j \in S$, i and j don't conflict

Goal: $\max \sum_{i \in S} v_i$



$p(1)=0$

$p(2)=0$

$p(3)=1$

$O = \{2\}$ $\quad \sum_{i \in O} v_i = 3$

O is the optimal solution

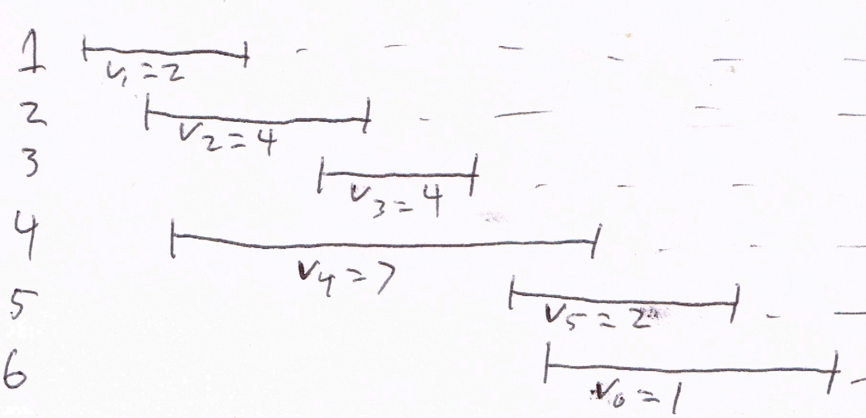
$G = \{1, 3\}$ $\quad \sum_{i \in G} v_i = 1 + 1 = 2$

---

w.l.o.g. assume $f_1 \leq f_2 \leq \dots \leq f_n$ (or sort them and relable)

Define: given i, $p(i)$ is the largest $j < i$ s.t. i doesn't conflict with j

compute $p(i)$ $1 \leq i \leq n$ (= 0 if no such $j < i$)



$p(1)=0$

$p(2)=0$

$p(3)=1$

$p(4)=0$

$p(5)=3$

$p(6)=3$

define: $\mathcal{O}_j$: Optimal solution for $\{1, \dots, j\}$

define: $OPT(j)$ = value for $\mathcal{O}_j = \sum_{i \in \mathcal{O}_j} v_i$

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

$\mathcal{O}_j = \{j\} \cup \mathcal{O}_{p(j)}$ $\quad OPT(j) = v_j + OPT(p(j))$

Case 2: $j \notin \mathcal{O}_j$

$OPT(j) = OPT(j-1)$

$\mathcal{O}_j = \mathcal{O}_{j-1}$

$\{1, \dots, j-1, j\}$ $\{1, \dots, j-1\}$

$OPT(j) = \max\left(v_j + OPT(p(j)),\ OPT(j-1)\right)$

given $OPT(1), OPT(2), \dots, OPT(j-1)$

we can compute $OPT(j)$

if $v_j + OPT(p(j)) > OPT(j-1) \Rightarrow j \in \mathcal{O}_j$

if $v_j + OPT(p(j)) < OPT(j-1) \Rightarrow j \notin \mathcal{O}_j$

---

if we compute $OPT(1), \dots, OPT(n)$

in $O(n)$ time, we have $\mathcal{O} = \mathcal{O}_n$ in $O(n)$ time.

$O(n \log n)$ with sorting