Alg 1:

$$T(n) = O(1) + T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) = O(1) + 2 \cdot T\left(\frac{n}{2}\right)$$

$$\Rightarrow O(n)$$

Alg 2:

$$T(n) = O(1) + T\left(\frac{n}{2}\right)$$

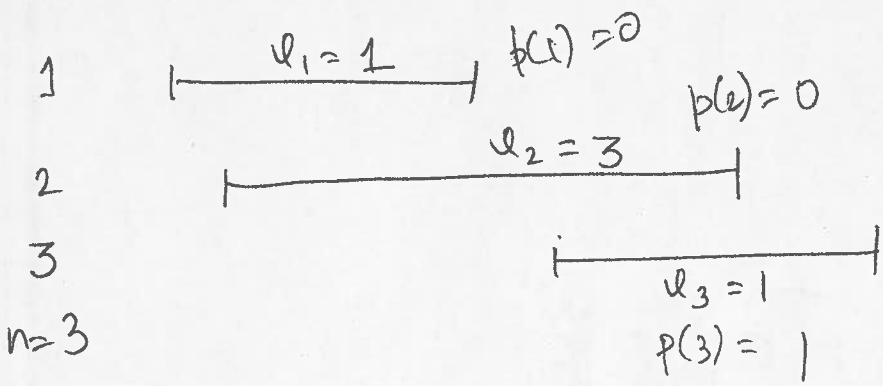
Weighted Interval Scheduling

I/p: n intervals: $s_i \rightarrow$ start time, $f_i \rightarrow$ finish time, $v_i \rightarrow$ value $\Big\}$ ith interval.

O/p: ~~A~~ valid schedule $S \subseteq [n] \overset{\text{def}}{=} \{1, \ldots, n\}$

$\hookrightarrow$ $\forall\, i \neq j \in S$, i & j do not conflict

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



Is $3 \in \mathcal{O}$?

Case 1: Yes

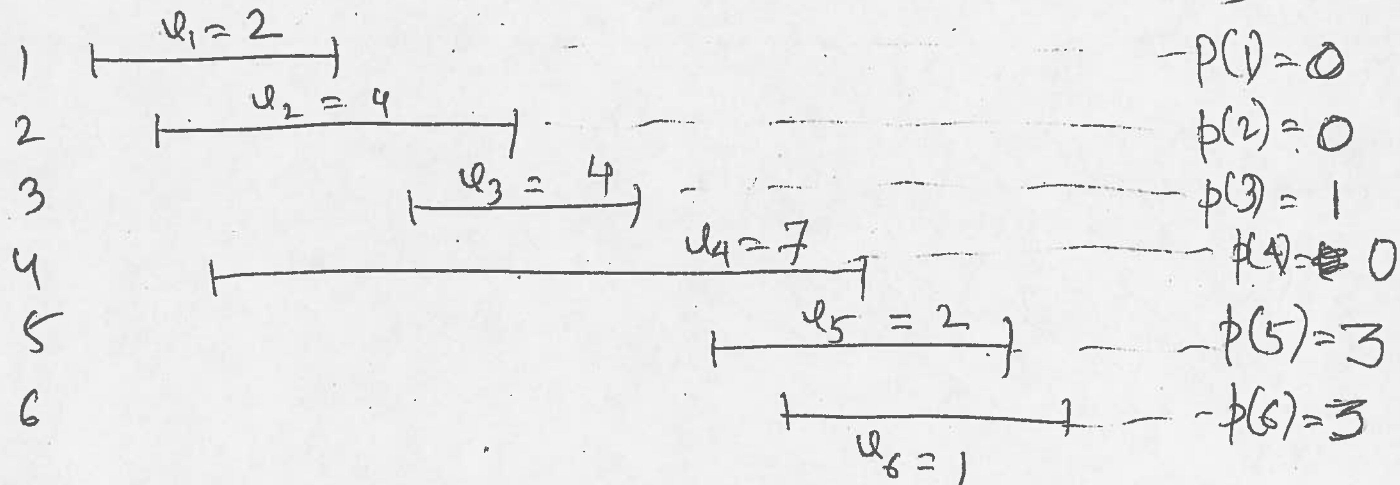
$\mathcal{O} = 3$; Look at $\{1, \ldots\}$

Case 2: No

$\mathcal{O} =$ Look at $\{2, 1, \ldots\}$

w.l.o.g assume $f_1 \le f_2 \le \dots \le f_n$

Def: Given i, $p(i)$ is the largest $j < i$ s.t. j doesn't conflict with i ($= 0$ if no such j $\exists$)



Ex: Algo to compute $p(i)$ $1 \le i \le n$.

Def: O_j optimal soln on $\{1, \dots, j\}$ $1 \le j \le n$

$\boxed{OPT(j)}$ = value of O_j = $\sum_{i \in O_j} v_i$

final $O = O_n$

$OPT(j)$ / O_j

Case 1: $j \in O_j$

j + remaining soln in $\{1, \dots, p(j)\}$

$O_j = j ; O_{p(j)}$ $\leftarrow$ By defn; $p(j)+1, \dots, j-1$ conflict w/ j.

Case 2: $j \notin O_j$

$O_j = O_{j-1}$