Weighted Interval scheduling.

- Given jobs $1 \ldots n$.
  Job j has a start time s_j
  end time f_j
  value v_j

- Goal: find a maximum weight subset of mutually compatible jobs.
  - Ex. $\{b, e, h\} \Rightarrow 3+3+3 = 9$
  - Ex $\{a, g\} \Rightarrow 6+4 = 10$

- Label jobs in order of finish time: $f_1 \leq f_2 \leq \ldots \leq f_n$

Define $p(j)$ = largest index $i < j$ s.t. job i is compatible w/ job j

$p(8) = 5 \quad p(7) = 3 \quad p(2) = 0$

Define $p(j)$ = largest index $i < j$ s.t. job i is compatible w/ job j

$p(8) = 5 \quad p(7) = 3 \quad p(2) = 0$

- Subproblem is to solve WIS for a smaller set of jobs, $\{1, \dots, i\}$. ~~(for exa~~
  (the first i jobs)
- Denote the max weight of a solution for the first i jobs as $OPT(i)$
- Trying to find $OPT(j)$, but you magically have the values for $OPT(i)$, $\forall i < j$
- What's changed/new in $OPT(j)$ that's not in $OPT(i)$?
  Job j!
- Job j is either in the OPT sol, or it is not.
  (inclusion/exclusion)

- If j is not included:

~~The OPT sol to j~~

The $OPT(j)$ solution is an optimal solution for the subproblem $\{1, \dots, j-1\}$.

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

- If j is included

The $OPT(j)$ solution contains an optimal solution to $OPT(p(j))$

$$OPT(j) = OPT(p(j)) + v_j$$

Dynamic Programming Formula/Equation.

$$\boxed{OPT(j) = \begin{cases} \max\left(OPT(j-1),\ OPT(p(j)) + v_j\right) & \text{if } j > 0 \\ 0 & \text{if } j = 0 \end{cases}}$$

Two strategies: memoization and bottom-up.

WIS-Bot-Up-DP $(n, s_1, \ldots, s_n, f_1, \ldots, f_n, v_1, \ldots v_n)$

$O(n\log n)$ Sort jobs by finish times

$O(n\log n)$ Compute $p(1), \ldots, p(n)$

$M(0) = 0$

for $j = 1$ to n {

DP table $\Theta(n)$ $\rightarrow$ $M(j) = \max(\underline{M(j-1)}, \underline{v_j} + \underline{M(p(j))})$ $\Theta(1)$

}

return $M(n)$

Runtime is $\Theta(n\log n)$