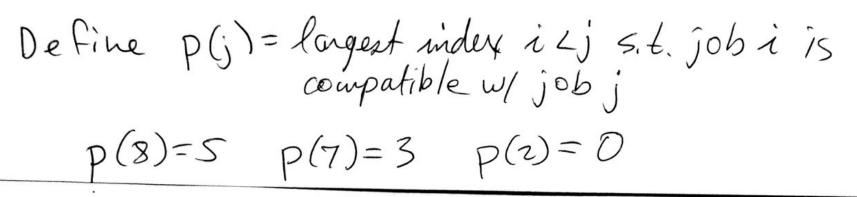
Weighted Interval scheduling. - 6 iven job 2 1... p. Jobj has a start time s; end time f; value y - 6 oal: find a maximum weight subset of mutually compatible jobs. -Ex. {b,e,h}=> 3+3+3=9 -Ex € a, g3 >> 6+4=10 - label jobz in order of finish time: f, &fz & ... &fn Define $p(j) = largest index i \(i \) s.t. job i is compatible \(w \) job j$ p(8)=5 p(7)=3 p(2)=0



- Subproblem is to solve WIS for a smaller set Of job2, & \SI,..., is. (for a smaller set)

 (the first i job2)
- Denote the max weight of a solution for the first ijobs as OPT (i)
- Trying to find OPT(i), but you magically have the values for OPT (i), tili
 - What's changed/new in OPT(j) that's not in OPT(i)?
 - Jobj is either in the OPT sol, or it is not. (inclusion/exclusion)

- elf j is not included: OF The OPT SOLO The OPT(j) solutions is an optimal solution for the subproblem (1,-j-13. OPT (j) = OPT (j-1) - Hj is included The OPT () solution contains an optimal OPT(j) = OPT(p(j)) + V_j pymamic Programming Formula/Equation solution to OPT (P(j)) $\left[OPT(j) = \sum \left(OPT(j-1) \right) OPT(p(j)) + V_j \right) \quad \text{if } j \geq 0 \\
 0 \quad \text{if } j = 0$ Two strategies: wemoisation and bottom-up.

WIS-Bot-Up-DP (n, si,.., sh, fy,.., fa, vi,..................) O(nlogn) Sortjobs by finish times O(nlogn) Compute P(i), ..., P(n) M(0)=0 for j=1 ton { >M(j)= max (M(j-1), v; + + M(p(j))) | O(1) return M(n)

Runtime is O(nlogn)