Last time:

- Shortest-paths (Bellman-Ford Alg)

Today:

- interval pricing

Example: Interval Pricing

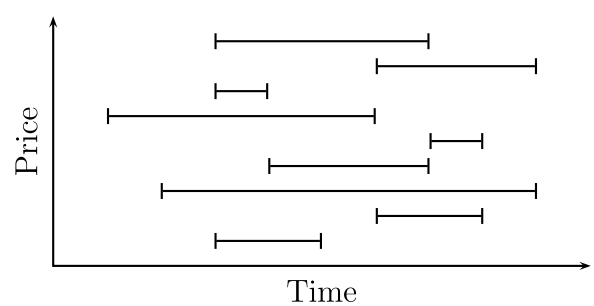
input:

- n customers $S = \{1, \dots, n\}$
- T days.
- i's ok days: $I_i = \{s_i, \dots, f_i\}$
- i's value: $v_i \in \{1, \dots, V\}$

output:

- prices $p[t]$ for day t.
- consumer i buys on day $t_i = \text{argmin}_{t \in I_i} p[t]$ if $p[t_i] \leq v_i$.
- revenue $= \sum_{i \text{ that buys}} p[t_i]$.
- goal: maximize revenue.

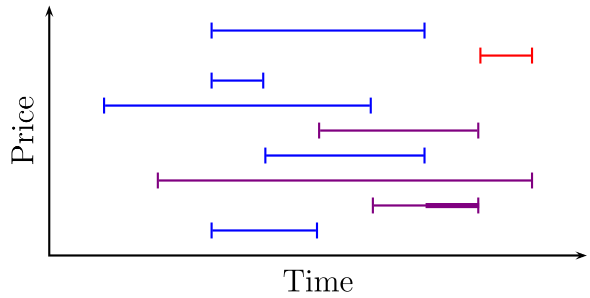
Example:



Question: What is "first decision we can make" to separate into subproblems?

Answer: day and price of smallest price.

Example:



Step I: identify subproblem in English

OPT(s, f, p)

$=$ "optimal revenue from intervals strictly between s and f with minimum price at least p"

Step II: write recurrence

OPT(s, f, p)

$$= \max_{s<t<f, q\geq p} \mathrm{Rev}(s,t,f,p) + \mathrm{OPT}(s,t,q) + \mathrm{OPT}(t,f,q).$$

Rev(s,t,f,p) = "the revenue from customers with interals within $[s,t]$ and overlapping t who are offered price p"

with

Step III: value of optimal solution

- optimal interval pricing $=$ OPT$(1, T, 0)$

Step IV: base case

- OPT$(s, s+1, p) = 0$.
- OPT$(s, t, P+1) = 0$.

Step V: iterative DP

(exercise)

Correctness

induction

Step VI: Runtime

- precompute Rev(t,p) in $O(TVn)$ time.
- size of table: $O(T^2V)$
- cost of combine: $O(TV)$.
- total: $O(T^3V^2)$ (assuming $n < T^2V$).

Note: without loss of generality T, V are $O(n)$ so runtime is $O(n^5)$

Note: can be improved to $O(n^4)$ with slightly better program.

Step VII: implementation

(exercise)