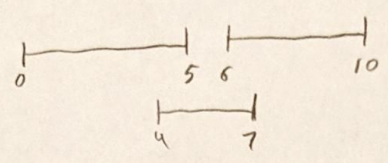


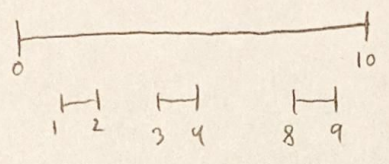
The scheduling problem is as follows. Given a set of tasks $S \subseteq \mathbb{R}^2$, where each task (s, t) has a start time s and end time t , what is the maximum number of tasks you can do? (tasks must not overlap)

How do we determine what tasks to perform?

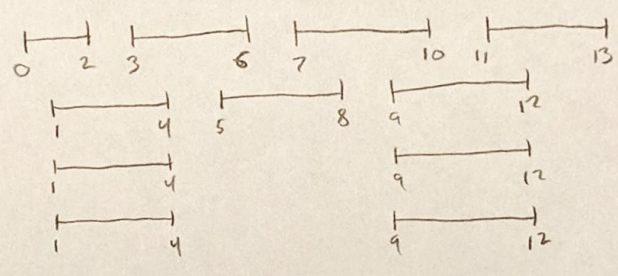
- Shortest Duration



- Next Available



- Least Conflicts



- DP

$$R(S) = 1 + \max_{t \in S} \left(R(\{t' \in S \mid t' \text{ ends before } t \text{ starts}\}) + R(\{t' \in S \mid t' \text{ starts after } t \text{ ends}\}) \right)$$

- Earliest Deadline (Greedy Solution)

Thm) Let $S \subseteq \mathbb{R}^2$ be a nonempty instance of the scheduling problem, and let $a = (t_s, t_f) \in S$ with $\forall a' = (t'_s, t'_f) \in S, t_f \leq t'_f$. Then a is included in some maximum size nonoverlapping subset of S .

pf) Let S, a be as stated.

Let $A \subseteq S$ be an optimal solution, and let $b = (r_s, r_f) \in A$ be such that $\forall b' = (r'_s, r'_f) \in A, r_s \leq r'_s$.

If $a = b$, we're done.

If $a \neq b$, then a satisfies $t_f \leq r_f$. Moreover, since no $b' \in A$ can overlap b , b' must start after b since b' finishes after b .

As such b' also starts after a . Thus $(A \setminus \{b\}) \cup \{a\}$ is also a maximum size non-overlapping subset of S , and it contains a . \square

SP(S)

Sort the tasks in S by their finish time.

Let T be an empty set of tasks.

Let $t = -\infty$

For $i = 1$ to $|S|$

 If $S_i.t_s \geq t$

 Add S_i to T

$t = S_i.t_f$

Return T

The runtime of SP is $O(n)$ + the runtime of whatever sort you choose to use. It is not unreasonable to require S to be sorted ahead of time, in which case you would not bother.