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

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

- Earliest Deadline (Greedy Solution)

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

\text{If } S_i.t_f \geq t

\quad \text{Add } S_i \text{ to } T

\quad t = S_i.t_f

\text{Return } T

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