Interval Scheduling

Most basic version:

Given a set of $N$ intervals:

![Diagram of intervals]

Each defined by a start time $S_i$ and finish time $F_i$.

(think: people that want to rent your house).

What is the maximum number of disjoint intervals you can pick?

That is, find

$$S \subseteq [N] \text{ maximizing } |S|$$

such that

$$[S_i, F_i) \cap [S_j, F_j) = \emptyset \quad \forall i, j \in S, \; i \neq j$$

The example has 4.
Many plausible greedy algorithms.

Shortest first?

Interval w/ fewest overlaps first?

First come first serve?

Earliest finish time?

**YES**

but needs a proof to show it's different from the above fallacious arguments.
Greedy Schedule \( \mathcal{I} \) :

\[ I = \{ (s, f) \mid s \in \mathbb{N} \} \]

Sort \( \mathcal{I} \) by increasing \( f \).

\[ S = \emptyset \]

\[ t = -\infty \]

For \( (s, f) \) in \( \mathcal{I} \):

if \( s < t \) : continue

\[ t = f \]

\[ S.\text{append}( (s, f) ) \]

Return \( S \).

**Theorem.** Greedy Schedule returns a valid set of intervals of maximum size.

**Proof.** Because \( s \leq f \) for every interval, \( t \) never decreases. Therefore \( t \) is always the largest \( f \) in \( S \). This ensures each new interval added to \( S \) does not overlap with any interval already in \( S \), so the returned \( S \) is valid.

We now prove by induction on \( n \) that the returned \( S \) has maximal size.

The base case of \( n = 0 \) is trivial. Now consider some \( n \geq 1 \), and assume the theorem holds for all \( I' \) with \( |I'| \leq n-1 \).

For any time \( t \), define \( I_+ = \{ (s, f) \in I \mid s > t \} \).

Notice that, if \( (s, f) \) has the earliest finish time in \( I_+ \), then

\[ \text{Greedy Schedule (I)} = I_+ \cup \text{Greedy Schedule (I_+)} \]
Let $\mathcal{S}$ be a maximum disjoint subset of $\mathcal{I}_F$ which we can write out in order as

$$\mathcal{S} = \{(s_1, f_1^*), (s_2, f_2^*), \ldots, (s_k^*, f_k^*)\}$$

For $k^* = |\mathcal{S}|$. Similarly, \textit{GreedySchedule} returns a $\mathcal{S}$ with

$$\mathcal{S} = \{(s_1, f_1), (s_2, f_2), \ldots, (s_k, f_k)\}$$

for $k := |\mathcal{S}|$. We claim $k = k^*$.

Since $f_1$ is the earliest possible finish time,

$$f_1 \leq f^*_1 \leq s_2^*.$$ 

Therefore $(s_2^*, f_2^*)$, $\ldots$, $(s_k^*, f_k^*)$ all lie in $\mathcal{I}_{f_1}$. Since these are disjoint, the optimal solution for $\mathcal{I}_{f_1}$ has at least $k^* - 1$ intervals.

By the inductive hypothesis, this means

$$|\text{GreedySchedule}(\mathcal{I}_{f_1})| \geq k^* - 1$$

so $k = |\text{GreedySchedule}(\mathcal{I})| \geq k^*$. Since $k^*$ is the maximum possible, $k = k^*$ as desired.

So: $O(n \log n)$ sort + $O(n)$ time.
Weighted interval Scheduling

Each interval now has a weight \( w_i \).

Maximize \( \sum_{i \in S} w_i \)

over sets of disjoint intervals.

Naive recursion:

\[
\text{Sched}(I)
\]

Let \( i = \text{first elt of I} \)

Return \( \min \) of

not chosen: \( \text{Sched}(I \setminus i) \),

chosen: \( \text{Sched}(I \setminus \frac{i}{3} \text{ or anything conflicting with } i \setminus 3) + w_i \).

Problem: \( 2^n \) possible inputs.

Solution: If I sorted by \( f_3 \), then only \( n+1 \) inputs ever happen:

(suffix of \( I \) sorted by \( s_1 \)).

\( \Rightarrow \) memoized time is \( n \cdot (\text{time per input}) \).

\[ n^2 \text{ naively} \]

\[ n \text{ more carefully} \]

\( \text{Sched} \) (index in \( I \), \( f \) or last chosen)