Greedy Algorithms

Make the "best" choice each time

Interval Scheduling

Input: intervals $(s_1, f_1), \dots, (s_n, f_n)$ (2n numbers $\{s_1, \dots, s_n, f_1, \dots, f_n\}$ different and > 0)

Output: choose max. # of pairwise-disjoint intervals.

"Scheduling" Problems.

DP approach: Imagine a directed graph $D = (V, A)$ s.t.
$V = \{\text{intervals}\}$ and
$(s_i, f_i) \to (s_j, f_j)$ when $f_i < s_j$.
D is DAG.
Then, Interval Scheduling $\equiv$ Longest Path!!
So, can be solved in time $O(|V| + |A|)$,
but $|A|$ can be $\Omega(|V|^2) = \Omega(n^2)$

IS-Greedy-?

$I \leftarrow \emptyset$.

While $\exists$ intervals that can be added to I

Add the "best one".

Output I.

Several natural choices for "best"

- "Shortest"
- "Earliest s_i / f_i"
- "Latest s_i / f_i"

Magically, will choose "earliest f_i"

IS-Greedy-Earliest f_i

$I = \emptyset$.

While $\exists$ intervals that can be added to I

Add "the one with earliest f_i".

Output I.

Running time: $O(n \log n)$ to sort s.t. $f_1 \leq \cdots \leq f_n$.
and $O(n)$ for the while loop.

```
I = ∅, f = 0
For i=1 to n
   If f < s_i
      I = I ∪ {(s_i, f_i)}
      f = f_i.
Return I
```

IS-Optimality

Correctness.
Let ALG be the set of intervals output by our algo.
Let OPT be the optimal solution.

First, ALG is a "valid" (i.e., all intervals are disjoint).
So $|ALG| \le |OPT|$.

WTS $|ALG| \ge |OPT|$.

Assume intervals are sorted: $f_1 \le \cdots \le f_n$

$ALG_i = ALG \cap \{(s_1,f_1), \ldots, (s_i,f_i)\}$.

$OPT_i = OPT \cap$ "

$f(ALG_i) = \max_{(s_j,f_j)\in ALG_i} f_j$. $f(OPT_i)$ similarly.

T_i or ALG$_i$ "dominates" OPT$_i$

Claim. $\forall 1 \le i \le n$, either $|ALG_i| > |OPT_i|$ or
$|ALG_i| = |OPT_i|$ and $f(ALG_i) \le f(OPT_i)$.

Proof. Induction on i. When $i=1$, easy to see.
When T_j holds for $j=1,\ldots,i-1$,

① $OPT_i \not\ni (s_i,f_i)$: $OPT_i = OPT_{i-1}$ and T_{i-1} is true, so is T_i.
(ALG$_{i-1}$ dominates $OPT_{i-1} = OPT_i$ and
ALG$_i$ dominates ALG$_{i-1}$)

② $OPT_i \ni (s_i, f_i)$: Let i' be largest index s.t. $f_{i'} < s_i$.

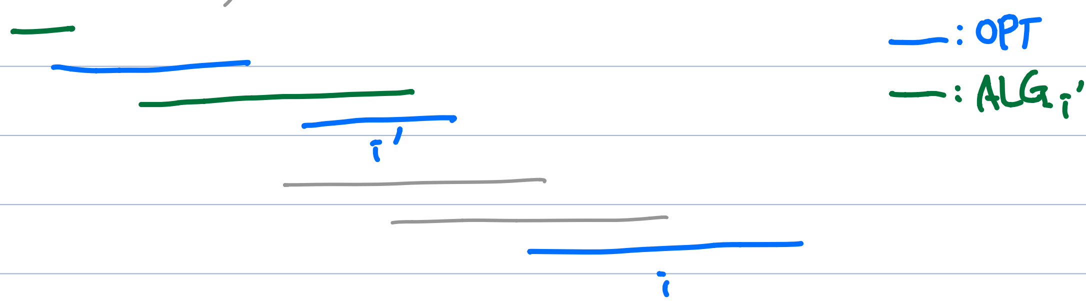
Then $OPT_i = \{(s_i, f_i)\} \cup OPT_{i'}$.

$T_{i'}$ is true, so is T_i.

$ALG_{i'}$ dominates $OPT_{i'}$

(i) If $|ALG_{i'}| \geq |OPT_{i'}| + 1 = |OPT_i|$, then $|ALG_{i'}|$ dominates OPT_i.

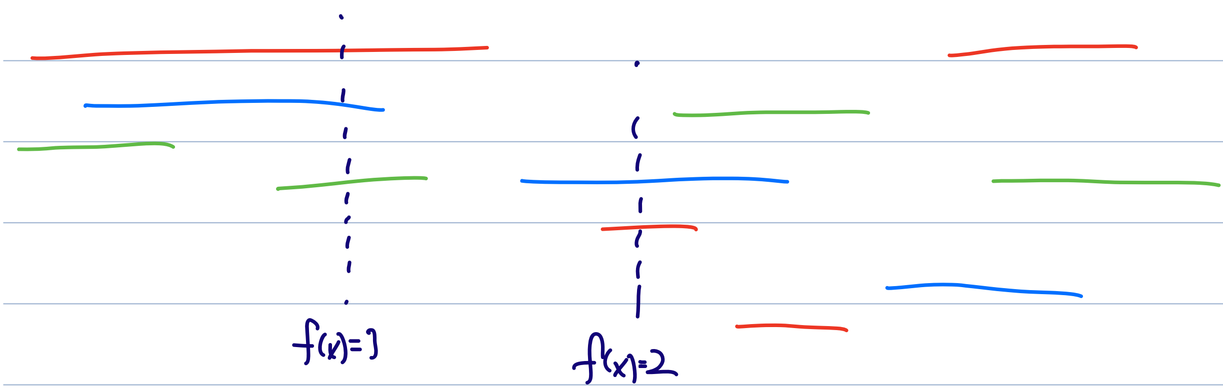
(ii) Otherwise,



After $ALG_{i'}$, algo. must have added either (s_i, f_i) or an earlier interval! $\square$

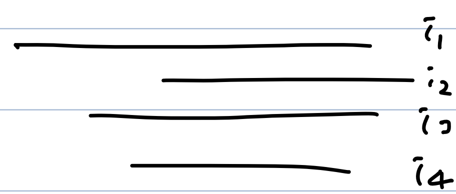
Minimizing number of rooms

New problem: Same input, but now schedule all intervals into smallest # of "rooms" s.t. intervals in each room are disjoint



Run previous algo repeatedly until all intervals are scheduled?
(# rooms) = (# times we run the alg)

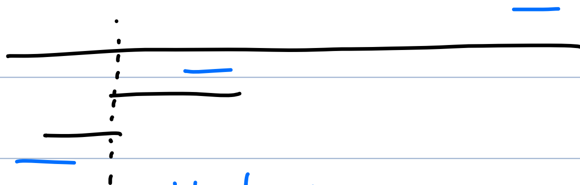
For $x \in \mathbb{R}$, let $f(x) = $ (# intervals containing x)
Optimal value $\geq \max_x f(x) =: k$
Can we find a solution with value $= k$?



Suppose $k=4$, take any 4 intervals that have a nonempty intersection

Is it possible that greedy algo chooses none of i_1, i_2, i_3, i_k?

- If NOT true, k is decreased by 1 in each room, and we'll be done using k rooms.



- But it is possible! ALG chooses —'s.

New Greedy (one room).

Sort intervals s.t. $s_1 \le \cdots \le s_n$.

$S \leftarrow \emptyset$.

For $i = 1, \dots, n$

  Add (s_i, f_i) to S if it fits.

Schedule intervals of S in one room.

Claim. Let $(s_{i_1}, f_{i_1}), \dots, (s_{i_k}, f_{i_k})$ be intervals s.t. $\bigcap_{j=1}^{k} (s_{i_j}, f_{i_j}) = [a, b]$.

k defined above!

Then the new greedy algo. takes at least one of them.

Pf. WLOG, let $s_{i_1} = a$. Assume for contradiction that the algorithm chose none of $(s_{i_1}, f_{i_1}), \dots, (s_{i_k}, f_{i_k})$.

Since (s_{i_1}, f_{i_1}) is not taken by the alg, alg must have taken some interval (s, f) s.t. $s < s_i$ and $f > s_i$.

But, it means $[s, f] \ni a$ s.t. $f(a) \ge k+1$ $\Rightarrow\Leftarrow$.