Oct 7

Interval Scheduling Problem

Input: n intervals — i^{th} interval: start time $s(i)$, finish time $f(i)$

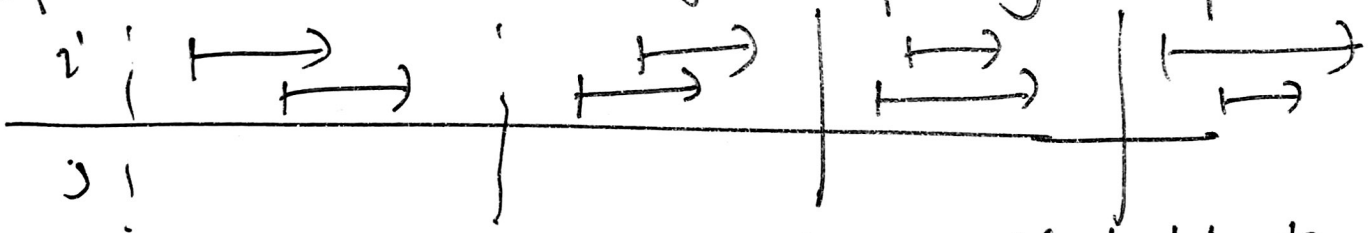
Convention: i^{th} interval $[s(i), f(i))$ $\quad [1,4) = \{1,2,3\}$

Output: A valid schedule with max # of intervals

Def: A schedule $S \subseteq [n]$

Def: A schedule S is valid if it has no conflicts.

Def: i & j are in conflict if they overlap.



Q: Given i & j how much time would it take to verify if they have a conflict.

A: $O(1)$ time. Check one of 4 conditions → each condition needs $O(1)$ comparisons

---

Obs: Sorting a valid schedule (⟼ ⟶ ⟼⟼ ⟶ .. by start times or finish time gives the same order.

Assume: The input is sorted by finish time

$f(1) \le f(2) \le \cdots \le f(n)$. [If not use any $O(n \log n)$ sorting algo]

Greedy Algo

0. $R = [n] \stackrel{\text{def}}{=} \{1, .., n\}$
1. $S = \{\}$
2. While $R \neq \{\}$
   (2.1) Let i be the smallest index of any interval in R
   (2.2) Add i to S
   (2.3) Delete i from R
   (2.4) Remove all j from R that conflicts with i.
3. Output $S^* = S$

THM 1: S^* is an optimal soln, i.e. has max # intervals among all valid schedules.

Ex1: Algo terminates

Ex2: S^* is a valid schedule.

Pf of correctness of greed algos → "Greedy stays ahead"
→ Exchange argument (later)

Let $\mathcal{O}$ be an optimal solution (i.e. among all possible valid schedules $\mathcal{O}$ has max # of intervals)

Ex: Convince yourself that such an $\mathcal{O}$ ∃.

Idea 1: Argue $S^* = \mathcal{O}$ (Problem: Not true if multiple optimal solns)

Idea 2: $|S^*| = |\mathcal{O}|$

THM 2: $|S^*| = |\mathcal{O}|$ ($\Rightarrow$ THM 1)