Oct 2

Interval Scheduling Problem

$[3,8) = \{3,4,5,6,7\}$

Input: n intervals : i^{th} interval $[s(i), f(i))$ ($s(i)$: start time; $f(i)$: finish time = $f(i)-1$)

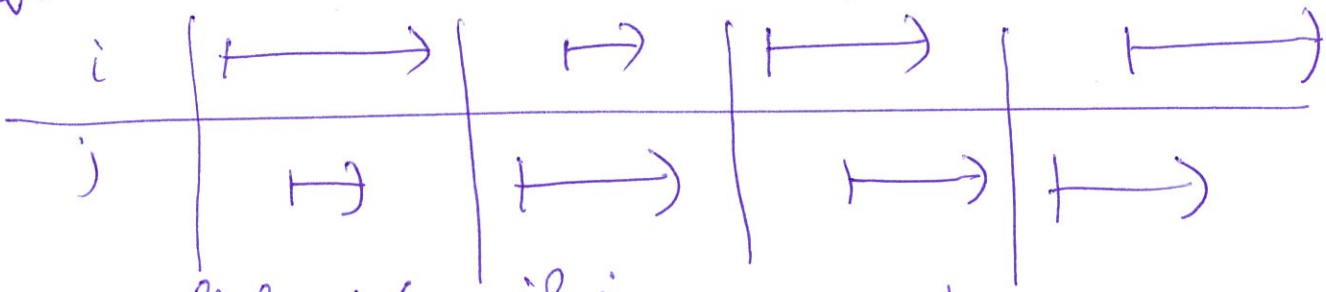
$1 \le i \le n$

Output: A valid schedule with max # of intervals in it.

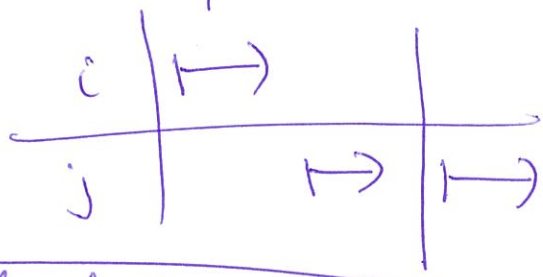
Def: A schedule $S \subseteq [n] \stackrel{def}{=} \{1, \ldots, n\}$

Def: A valid schedule has no conflicts

Pf: Interval i & j conflict if they overlap (as intervals)



$\Rightarrow$ no conflict b/w i & j



Claim: A valid schedule sorted by either start time or finish time, you get the sorted order.

Assume: Input interval are sorted in non-decreasing order of finish time

$f(1) \le f(2) \le \ldots \le f(n)$

↳ If not: in $O(n \log n)$ time sort the intervals.

Greedy Algo

0. $R \leftarrow [n]$
1. $S \leftarrow \phi$
2. While $R \neq \phi$
   (2.1) Let i be the smallest index in R (ok since $f(1) \leq \dots \leq f(n)$)
   (2.2) Add i to S
   (2.3) Delete all j (from R) that conflict with i (i conflicts with j itself)
3. Return $S^* \leftarrow S$

Thm: S^* is an optimal solution (for all inputs, among all possible valid schedules S for that input, S^* has the max # intervals $|S^*| \geq |S|$)

Pf of correctness
- → Greedy stays ahead (next)
- → Exchange argument (min max lateness → Sec 4.2)

Greedy stays ahead

Order S^*

$i_1, \dots, i_r, \dots, i_m$

$j_1, \dots, j_r, \dots, j_k$

S^* is doing "just as well" as Θ