Oct 4

Interval Scheduling Problem

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

Input: n intervals : i^{th} interval $[s(i), f(i))$

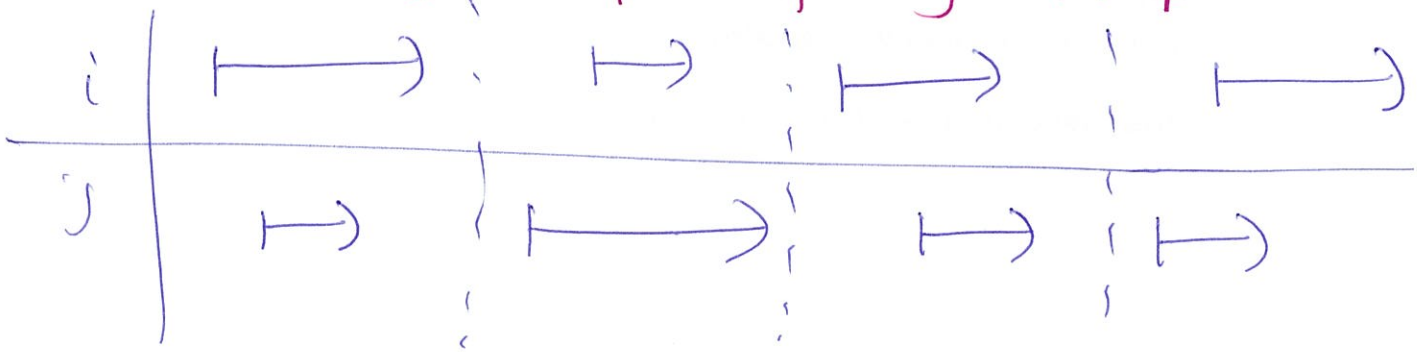
($f(i)-1$ is finish time; $s(i)$ is start time)

Output: A valid schedule with max # intervals in overall valid schedul

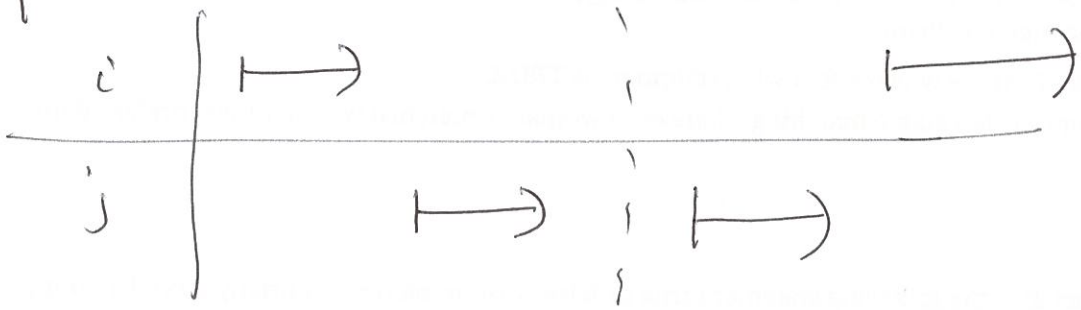
Def: A schedule $S \subseteq [n] \stackrel{def}{=} \{1, \ldots, n\}$

Def: A valid schedule S has no conflicts

Def: interval i & j conflict if they overlap



$\Rightarrow$ no conflict



Obs: A valid schedule ⟶ ⟶ ⟶ ⟶ sorted by start / finish time gives the same order.

Assume: Input intervals are sorted (→ increasing order) by finish time

$\Rightarrow f(1) \le f(2) \le f(3) \le \cdots \le f(n)$

↑ If not, sort in $O(n \log n)$ time

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
   - (2.2) Add i to S
   - (2.3) Remove i from R
   - (2.4) Delete all $j \in R$ that conflict with i
3. Return $S^* \leftarrow S$

THM: S^* is an optimal solution.

↳ ie $\forall$ inputs, among all possible valid schedules for the input, S^* has the max. # of intervals.

Ex 1: Algo terminates

Ex 2: S^* is a valid schedule

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