Sep 30

Interval Scheduling Problem

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

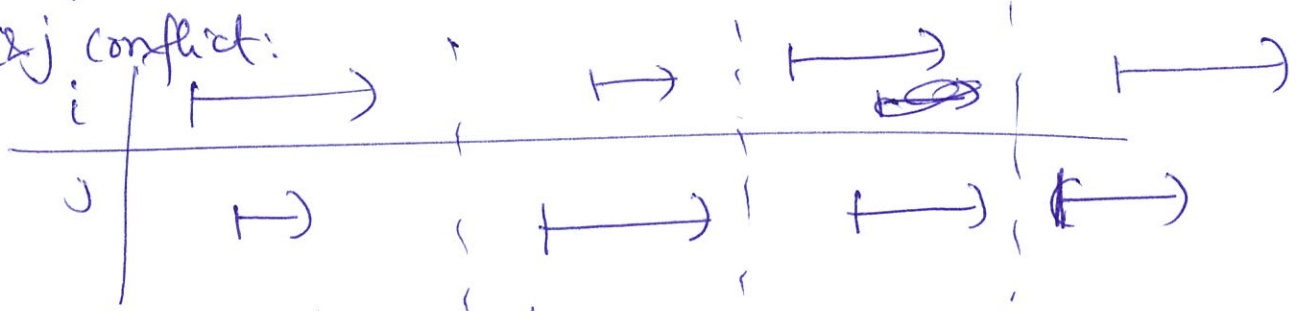
Input: n intervals : i^{th} interval $[s(i), f(i))$

Output: A valid schedule with max # intervals

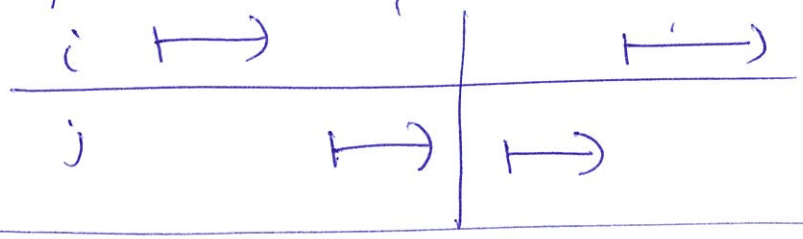
Def: A schedule $S \subseteq [n]$ $(= \{1, \cdots, n\})$

Def: A valid schedule S has no conflicts

Def: i & j conflict:



no conflict



Obs: A valid schedule sorted by start or finish time gives same order

Assume: Input intervals are sorted by finish time

$(f(1) \leq f(2) \leq \cdots \leq f(n))$ ⟸

↳ If not sort in $O(n \log n)$ time

Greedy Algo

0. $R = [n]$
1. $S = \phi$
2. While $R \neq \phi$
   - (2.1) Let i be the smallest index in R
   - (2.2) Add i to S
   - (2.3) Remove all i from R
   - (2.4) Delete j from R that conflict with i
3. Return $S^* = S$

Thm 1: S^* is an optimal solution
↳ $\forall$ inputs, among all possible valid schedules for given input, S^* has max # jobs.

Ex1: Algo terminates

Ex2: S^* is a valid schedule

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

Let $\mathcal{O}$ be an optimal solution

Ex3: Convince yourself that such an $\mathcal{O}$ $\exists$.

Idea: $S^* = \mathcal{O}$ ⟼ $\mathcal{O}$
↳problem: Can have ⟼ S^* > 1 optimal soln.

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