Oct 6, 2014

# Interval Scheduling Problem

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

"open interval" $[s(i), f(i))$

$s(i) = 1$ , $f(i) = 4$

$[1, 4) = \{1, 2, 3\}$

before $f(i)$

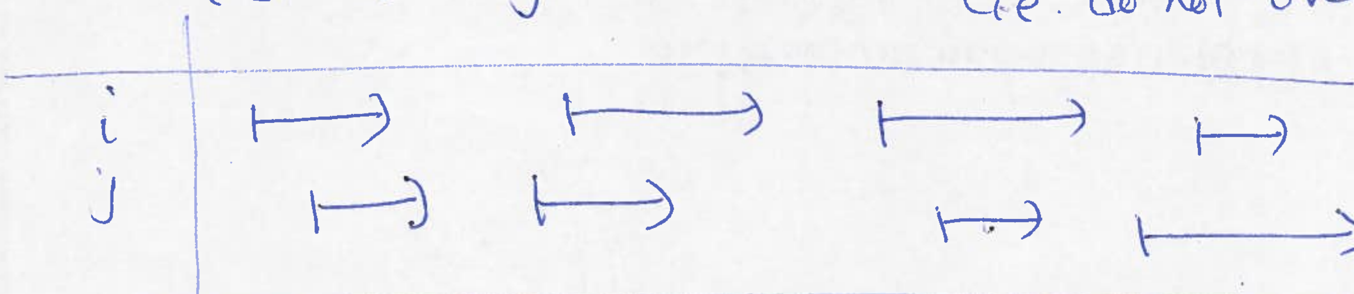
**Output:** A valid schedule w/o conflicts with max # intervals.

---

**Def:** Schedule: $S \subseteq [n]$

**Def:** Valid schedules: has no conflict.

i.e. $\forall\ i \neq j \in S$ , $i$ & $j$ do not conflict.
(i.e. do not overlap)




---

Q: How much time needed to check if $i$ & $j$ conflict.

A: $O(1)$ time : check if one of the 4 situations above ( $O(1)$ many comparisons )

**Obs:** Sorting a valid schedule by start or finish time results in the same order.

**Assume:** All input intervals sorted by ~~start tim~~ finish time

**Greedy Algo:**

1. $R \leftarrow [n]$ $= \{1, \ldots, n\}$ (all intervals)
2. $A \leftarrow \emptyset$
3. While $R \neq \emptyset$
   (3.1) Let $i \in R$ be the interval with the earliest finish time in $R$.
   (3.2) Add $i$ to $A$
   (3.3) Delete all $j$ from $R$ that conflict with $i$. (Note: $i$ conflicts with $i$)
4. Output $A$

---

EX: Algo always terminates.

**THM:** $A$ is a valid schedule with max # intervals.

---

Proof of correctness (for greedy algos)
- → Greedy stays ahead (now)
- → Exchange argument (next)

---

**Obs:** $A$ is a valid schedule (Ex.)

**To prove:** $A$ has max # intervals

**Analysis idea:** Assume $O$ is an optimal valid schedule.

**Idea 1:** Argue $A = O$ (Problem: More than one optimal sol'n possible)

**Idea 2:** Argue $|A| = |O|$

Def: $\mathcal{O}$ is a valid schedule with max # intervals.

THM: $|A| = |\mathcal{O}|$

Assume $A$ & $\mathcal{O}$ are sorted by finish time

$A = \{i_1, \ldots, i_k\}$ $\quad \mathcal{O} = \{j_1, \ldots, j_m\}$

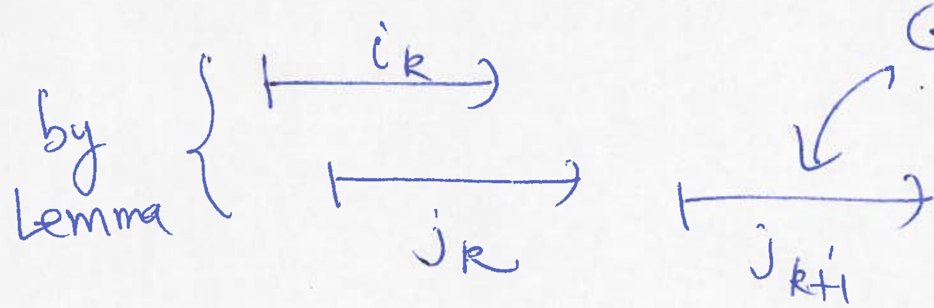
To prove: $k = m$.

Q: $k \le m$ ? [Otherwise $\mathcal{O}$ is not opt.]

Lemma 1: $\forall \; 1 \le l \le k$

$$f(i_l) \le f(j_l) \quad \text{(Assume true)}.$$

Pf of THM: For contradiction assume $k < m$




Greedy could have picked $j_{k+1}$ $\Rightarrow$ algo did not terminate. (as $m > k$)