Interval Scheduling Problem

Bakh Khoussainov

We have n requests:

$1, 2, \ldots, n$.

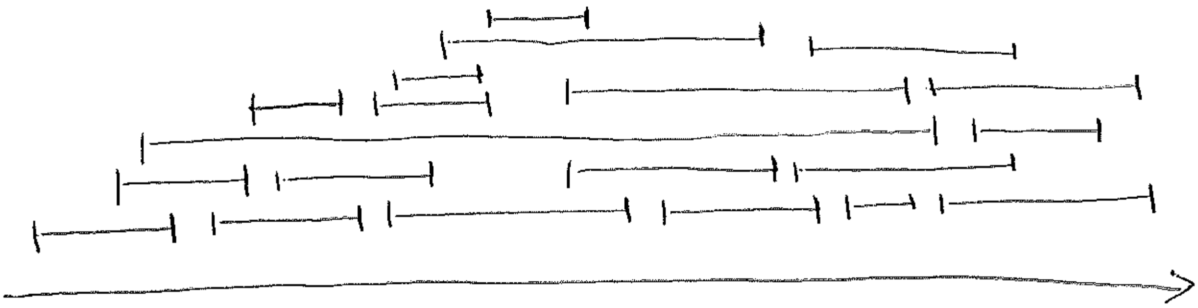
The starting and the finishing times of the i^{th} request are:

$s(i)$, $f(i)$.

A set of requests is <u>compatible</u> if no two of them overlap in time.

Goal: maximize the number of compatible requests.

Example:



A compatible set with the largest number of requests is called an optimal solution.

What is the size of an optimal solution in the example above?

An algorithm that finds an optimal solution is this.

(1) Initially $R = \{1, 2, \ldots, n\}$, $A = \phi$

(2) While $R \neq \phi$, select $i \in R$ with smallest finishing time.

Add i to A.

Delete all requests from R not compatible with i.

(3) Return A.

Properties of A.

Property 1. A is a compatible set of requests.

Indeed, the algorithm guarantees this.

Let $i_1, i_2, \ldots, i_k$ be all the requests in order they were added to A. So we have

$$s(i_1) < f(i_1) \leq s(i_2) < f(i_2) \leq \ldots \leq s(i_k) < f(i_k)$$

Let $\mathcal{O}$ be an optimal solution. List all requests in $\mathcal{O}$:

$$j_1, j_2, \ldots, j_m.$$

Goal: want to show
$$k = m.$$

Property 2. For each index r we have
$$f(i_r) \leq f(j_r).$$

Proof. When $r=1$ then

$$f(i_1) \leq f(j_1).$$

Assume

$$f(i_{r-1}) \leq f(j_{r-1}).$$

We know that

$$f(j_{r-1}) \leq s(j_r).$$

Thus, $f(i_{r-1}) \leq f(j_{r-1}) \leq s(j_r)$.

So, $j_r \in R$ when the algorithm selects i_r. By the selection rule $f(i_r) \leq f(j_r)$.

Property 3. A is optimal, that is $m = k$.

Assume $k < m$. Then $f(i_k) \leq f(j_k)$ and $f(j_k) \leq s(j_{k+1})$. Hence, after selecting i_k, we still have $R \neq \emptyset$ since $j_{k+1} \in R$.

Hence the algorithm must put a request i_{k+1} into A. Contradiction.