Interval Partitioning Problem

Bakh Khoussainov

Interval partitioning problem.

We are given n requests:

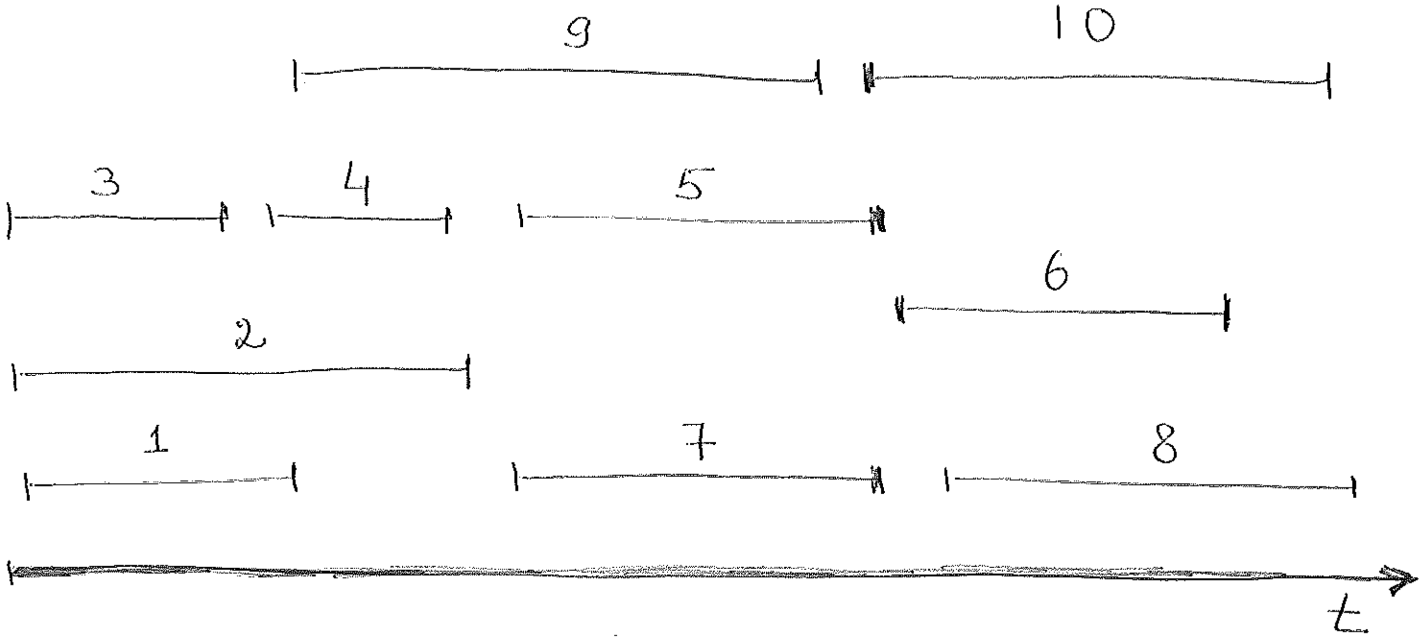
$1, 2, \ldots, n$.

Each request i starts at time $s(i)$ AND finishes at $f(i)$.

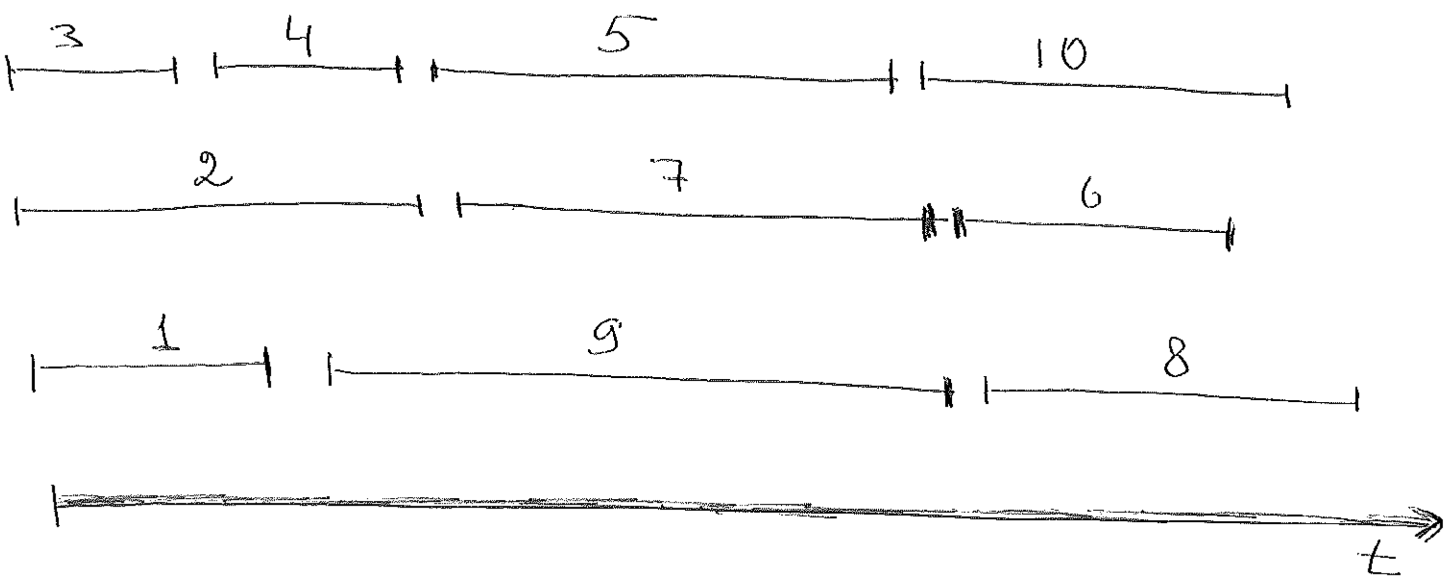
Suppose we have resources available to satisfy each request.

Goal: Schedule all requests using as few resources as possible.

Example



solution



The depth of a set of intervals is the maximum number that pass over any single point on the time line.

Property. The number of resources needed is at least the depth of the set of intervals.

We order requests by their starting time.

Let

$$l_1, l_2, \ldots, l_n$$

time intervals of these ordered requests.

Here is the algorithm that solves the interval partitioning problem.

For $j = 1, 2, \ldots, n$

For each l_i, $i < j$, if l_i overlaps l_j then exclude the label for l_i from consideration for l_j.

Assign the first nonexcluded label to l_j.

Note: our labels are $1, 2, 3, \ldots$

Property 1. Each interval is labeled.

Indeed, this is what the algorithm does.

Property 2. If t is a label of an interval then $t \le d$.

Consider interval l_j. Assume there are s intervals l_i, $i<j$, that overlap with l_j. So $s \le d-1$. Hence, $t \le d$.

Property 3. No two overlapping intervals have the same label.

Let l_j and l_i be overlapping intervals, $i<j$. Then the algorithm excludes the label of l_i from consideration for l_j.

All these prove that the algorithm is correct.