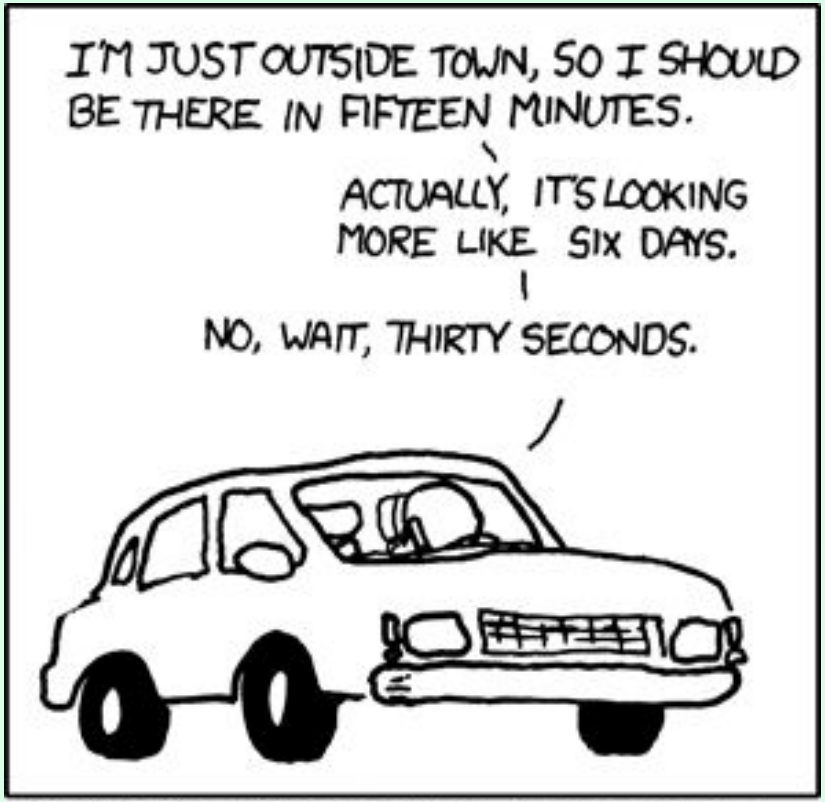
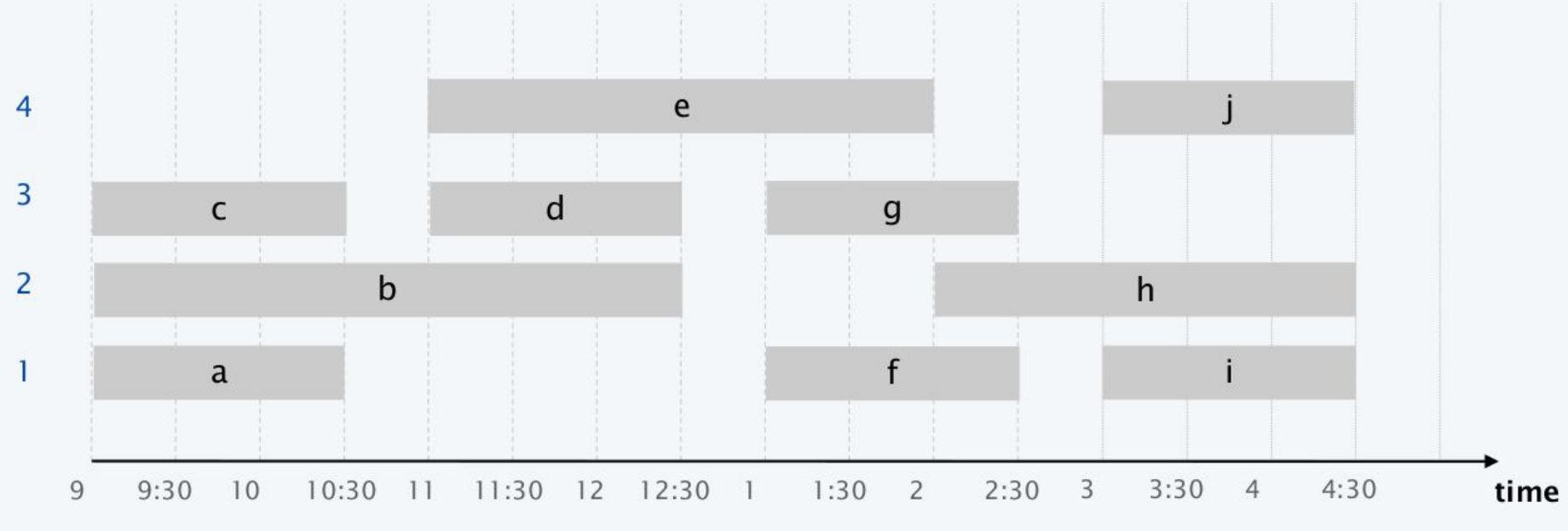
CS 310



THE AUTHOR OF THE WINDOWS FILE COPY DIALOG VISITS SOME FRIENDS.

Interval Partitioning Problem

Ex. This schedule uses 4 classrooms to schedule 10 lectures.

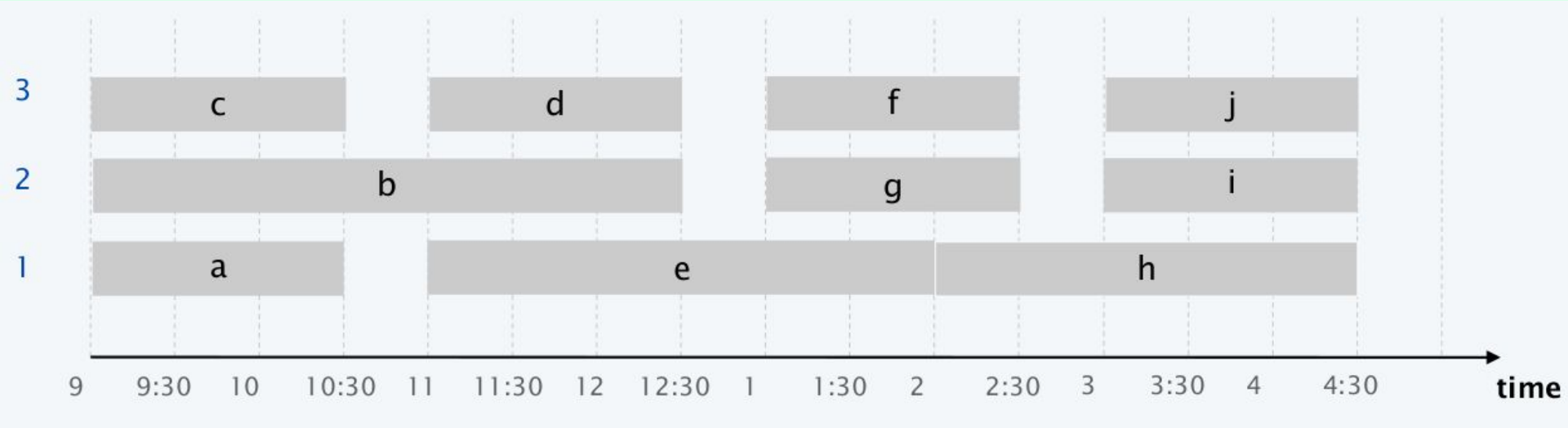


Interval Partitioning Problem

- Multiple resources are available.
- Schedule all request intervals using as few resources as possible.

Example: classrooms and courses

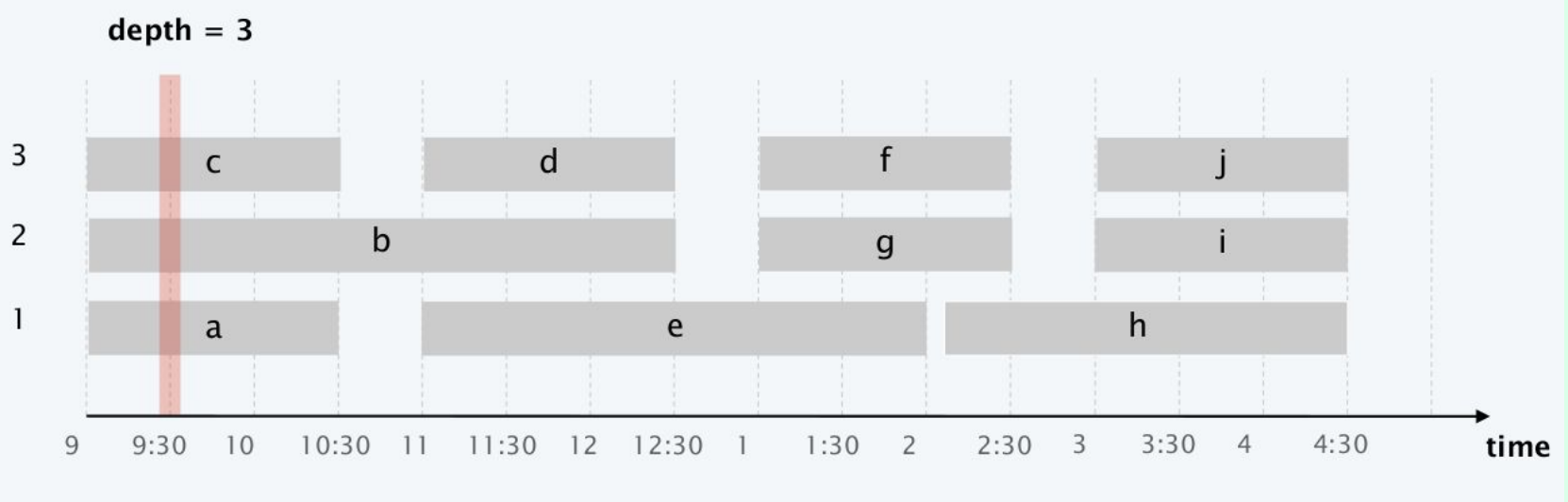
Interval Partitioning Problem



Can we schedule all lectures in 2 classrooms?

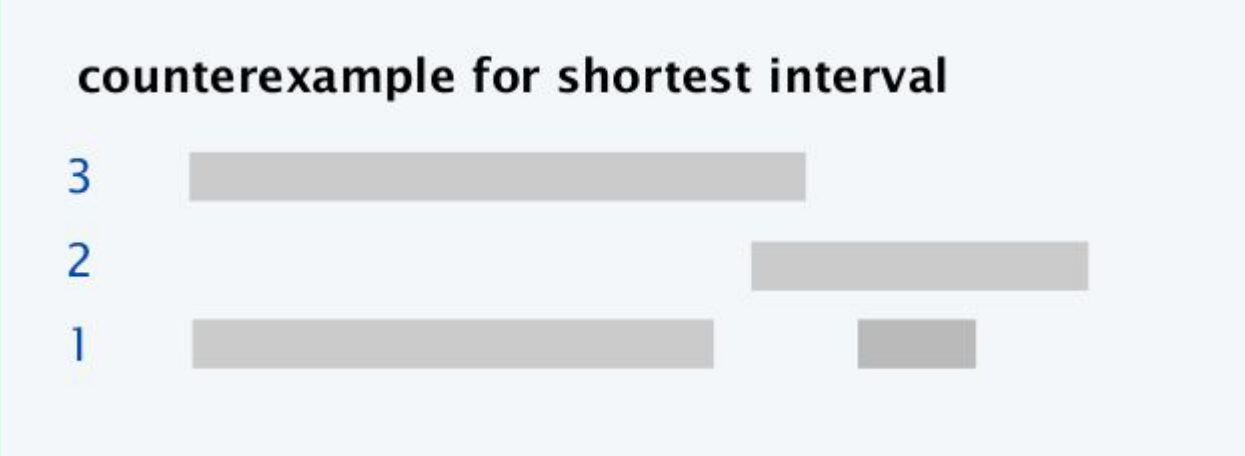
Interval Partitioning Problem

Def. The depth of a set of intervals is the maximum number that pass over a single point on the timeline.

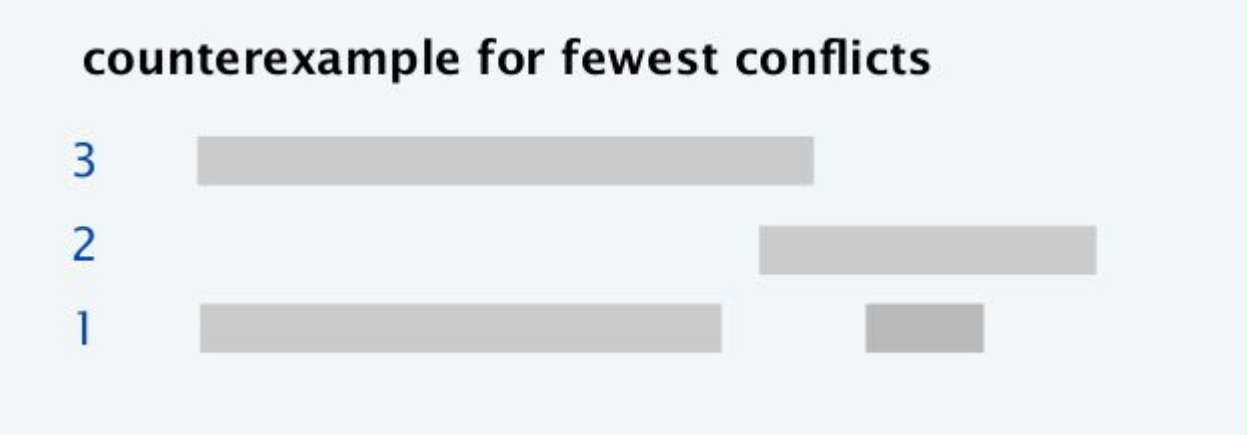


Any two lectures that overlap in time must be scheduled in different classrooms.

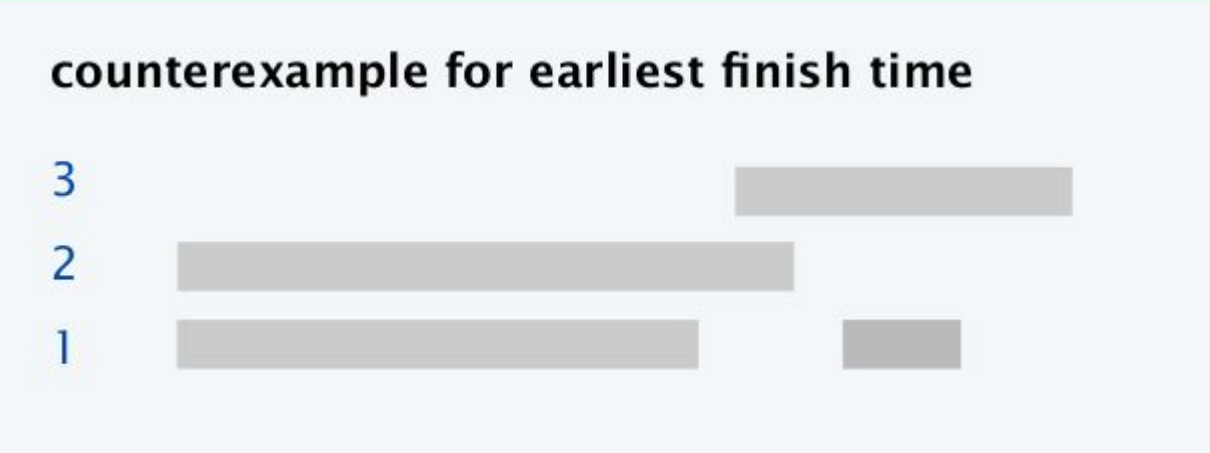
Interval Partitioning Problem



Interval Partitioning Problem



Interval Partitioning Problem



EARLIEST-START-TIME-FIRST $(n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n)$

SORT lectures by start time so that $s_1 \leq s_2 \leq \ldots \leq s_n$.

$d \leftarrow 0$ ← number of allocated classrooms

FOR $j = 1$ TO n

  IF lecture j is compatible with some classroom

    Schedule lecture j in any such classroom k.

  ELSE

    Allocate a new classroom $d + 1$.

    Schedule lecture j in classroom $d + 1$.

    $d \leftarrow d + 1$

RETURN schedule.

Complexity?

EARLIEST-START-TIME-FIRST $(n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n)$

SORT lectures by start time so that $s_1 \leq s_2 \leq \ldots \leq s_n$.

$d \leftarrow 0$ ← number of allocated classrooms

FOR $j = 1$ TO n

  IF lecture j is compatible with some classroom

    Schedule lecture j in any such classroom k.

  ELSE

    Allocate a new classroom $d + 1$.

    Schedule lecture j in classroom $d + 1$.

    $d \leftarrow d + 1$

RETURN schedule.

Store classrooms in a priority queue (key = finish time of its last lecture).

To determine whether lecture j is compatible with some classroom, compare s_j to key of min classroom k in priority queue.

- To add lecture j to classroom k, increase key of classroom k to f_j .

EARLIEST-START-TIME-FIRST $(n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n)$

SORT lectures by start time so that $s_1 \leq s_2 \leq \ldots \leq s_n$.

$d \leftarrow 0$ ← number of allocated classrooms

FOR $j = 1$ TO n

  IF lecture j is compatible with some classroom

    Schedule lecture j in any such classroom k.

  ELSE

    Allocate a new classroom $d + 1$.

    Schedule lecture j in classroom $d + 1$.

    $d \leftarrow d + 1$

RETURN schedule.

Reference reading:

Algorithm Design by Tardos et. al.
Interval Scheduling: §4.1