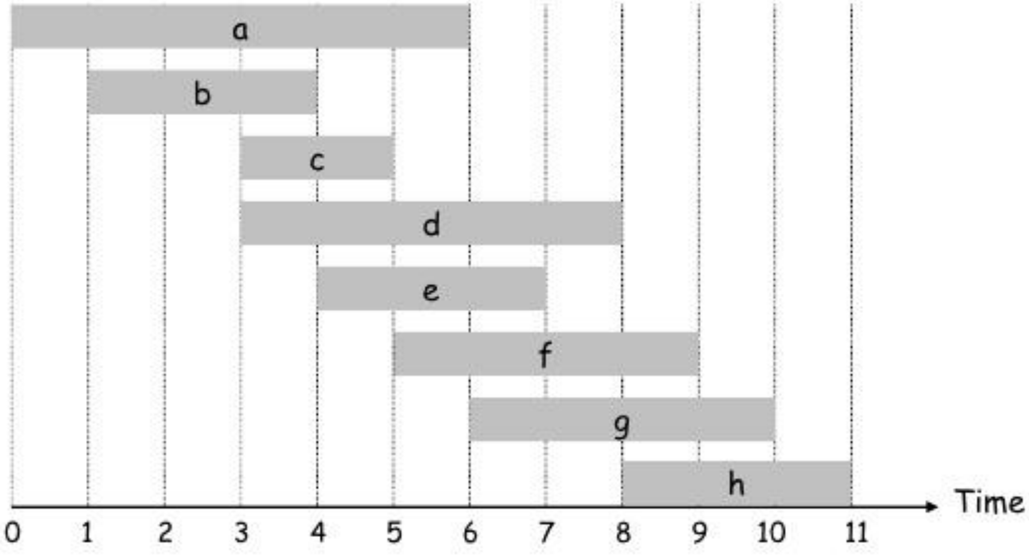
CS 310


HOCUS-POCUS, ABRACADABRA!
I COMMAND MY HOMEWORK TO DO ITSELF! HOMEWORK, BE DONE!
WATERSON
FLIP FLIP FLIP
RATS.
4-25

Interval Scheduling

Interval scheduling.

- Job j starts at s_j and finishes at f_j.
- Two jobs compatible if they don't overlap.
- Goal: find maximum subset of mutually compatible jobs.

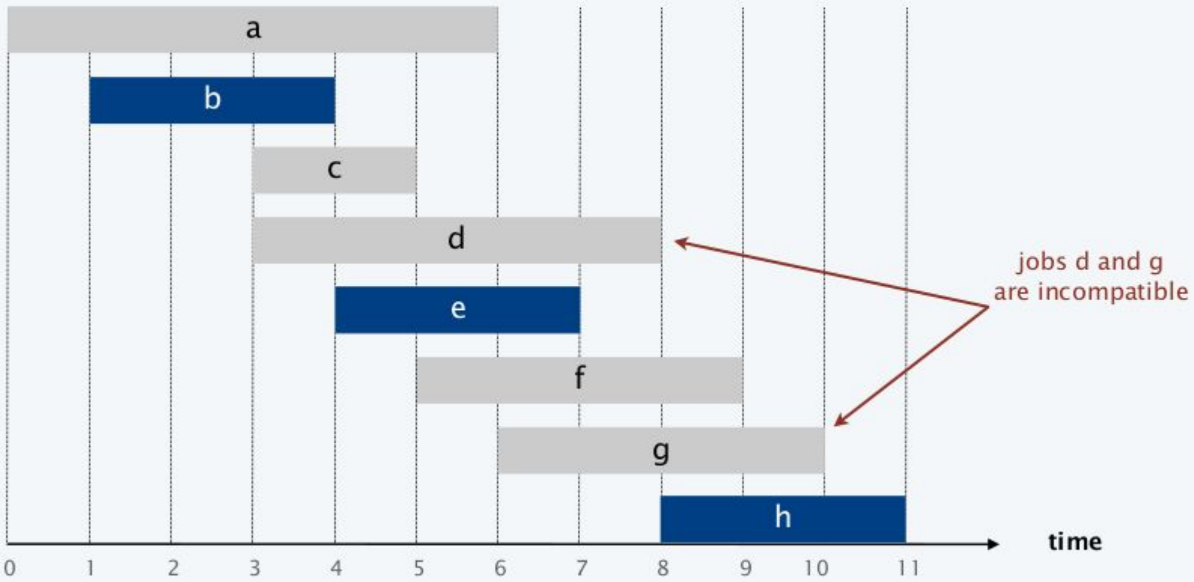


6

One classroom analogy

Interval scheduling

- Job j starts at s_j and finishes at f_j.
- Two jobs compatible if they don't overlap.
- Goal: find maximum subset of mutually compatible jobs.



Interval scheduling: earliest-finish-time-first algorithm

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

SORT jobs by finish time so that $f_1 \leq f_2 \leq \ldots \leq f_n$

$A \leftarrow \phi$ ← set of jobs selected

FOR $j = 1$ TO n

  IF job j is compatible with A

    $A \leftarrow A \cup \{ j \}$

RETURN A

Greedy Algorithms

- Short Sighted greed : not concerned about the rest of the intervals
- There can be more than one correct optimal solutions

Greedy Algorithms

- Optimal substructure property

Interval Scheduling - Earliest-finish-time-first

Is this algorithm optimal?

Interval Scheduling - Earliest-finish-time-first (EFTF)

Let OPT = Optimal set of intervals

A = Set of intervals by EFTF

Interval Scheduling - Earliest-finish-time-first (EFTF)

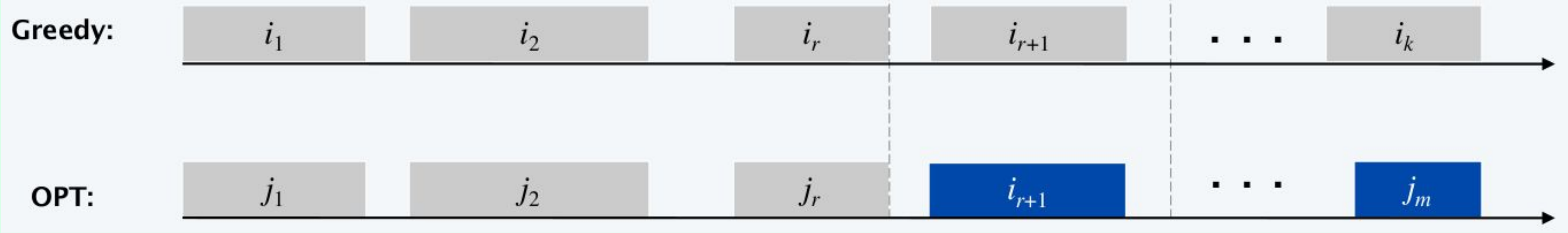
Let OPT = Optimal set of intervals

A = Set of intervals by EFTF

Show:

$|OPT| = |A|$

Interval Scheduling - Earliest-finish-time-first (EFTF)



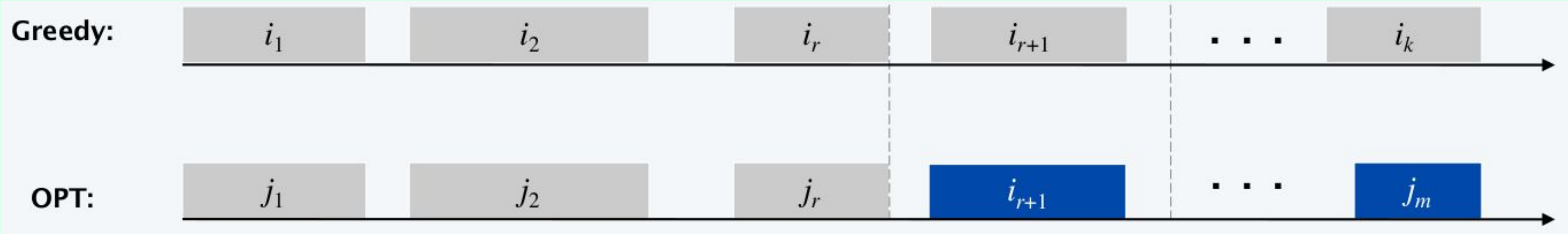
Show:

$k = m$

Interval Scheduling - Earliest-finish-time-first (EFTF)

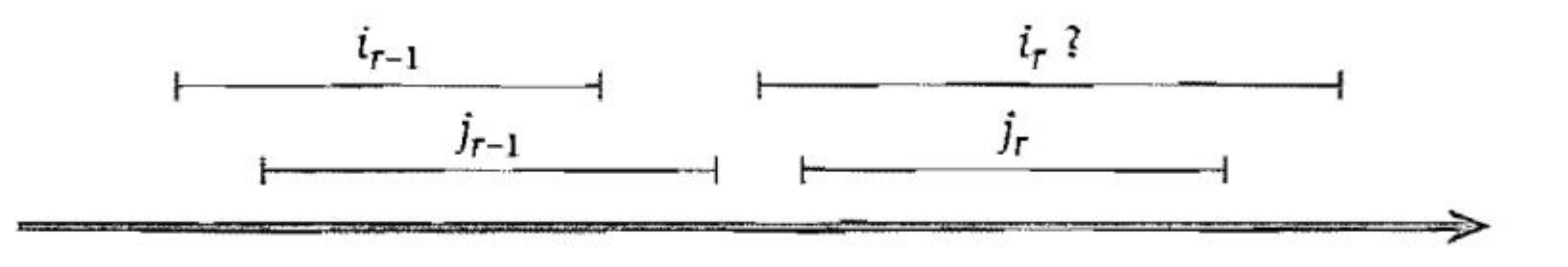
EFTF Intuition: Free up resource as soon as possible.

EFTF "Stays Ahead": Each of the intervals in A finishes as soon as the corresponding interval in OPT.



Interval Scheduling - Earliest-finish-time-first (EFTF)

Show: For all indices $r \leq k$ we have finish(i_r) $\leq$ finish(j_r)

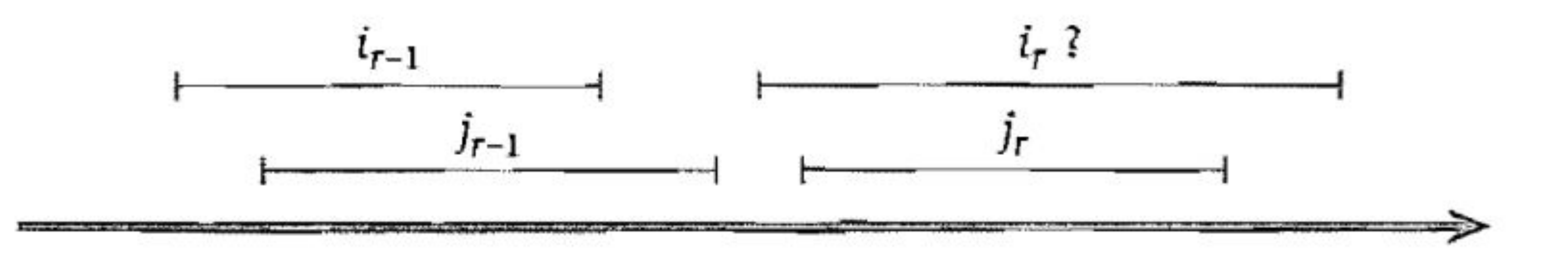


Interval Scheduling - Earliest-finish-time-first (EFTF)

Show: For all indices $r \leq k$ we have $finish(i_r) \leq finish(j_r)$

<u>Proof by Induction</u>

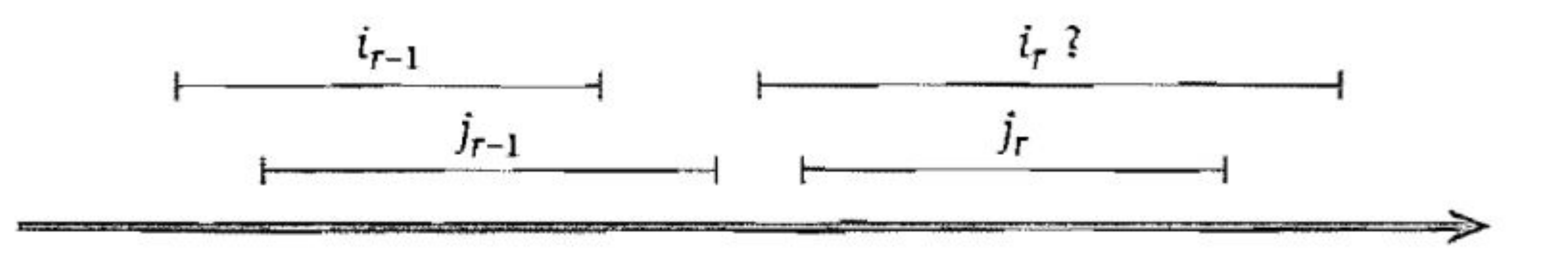
For r =1



Interval Scheduling - Earliest-finish-time-first (EFTF)

Show: For all indices $r \leq k$ we have $\text{finish}(i_r) \leq \text{finish}(j_r)$

<u>Inductive Hypothesis:</u> $\text{finish}(i_{r-1}) \leq \text{finish}(j_{r-1})$

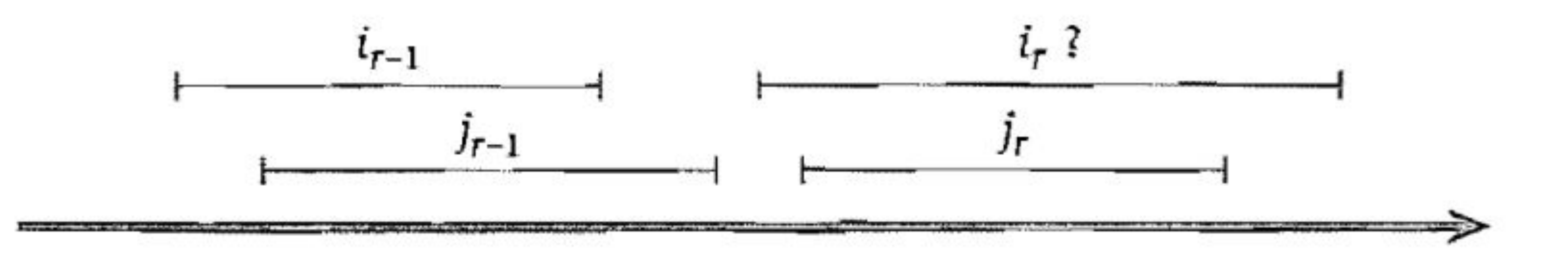


Interval Scheduling - Earliest-finish-time-first (EFTF)

Show: For all indices $r \leq k$ we have $\text{finish}(i_r) \leq \text{finish}(j_r)$

<u>Inductive Hypothesis:</u> $\text{finish}(i_{r-1}) \leq \text{finish}(j_{r-1})$

$\text{finish}(j_{r-1}) \leq \text{start}(j_r)$



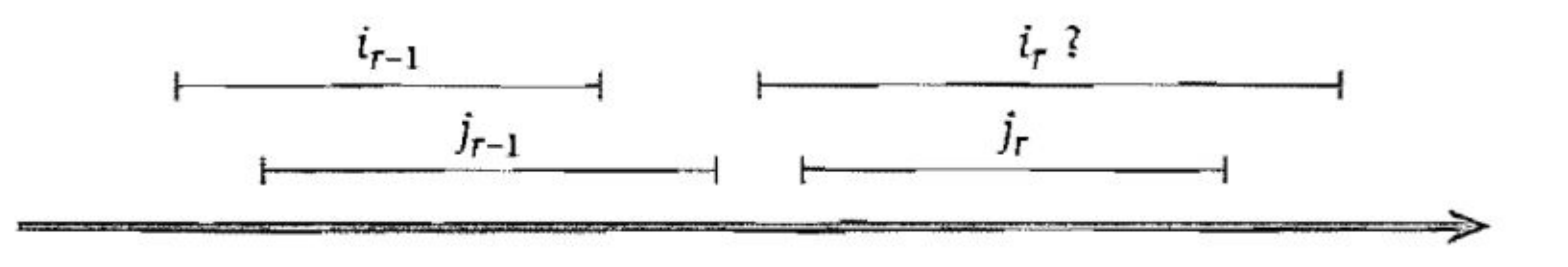
Interval Scheduling - Earliest-finish-time-first (EFTF)

Show: For all indices $r \leq k$ we have $\text{finish}(i_r) \leq \text{finish}(j_r)$

<u>Inductive Hypothesis:</u> $\text{finish}(i_{r-1}) \leq \text{finish}(j_{r-1})$

$\text{finish}(j_{r-1}) \leq \text{start}(j_r)$

$\text{finish}(i_{r-1}) \leq \text{start}(j_r)$



Interval Scheduling - Earliest-finish-time-first (EFTF)

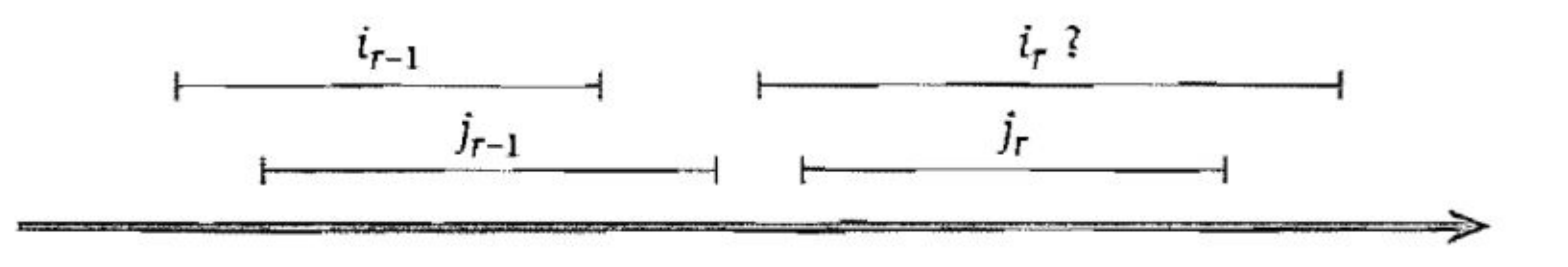
Show: For all indices $r \leq k$ we have $\text{finish}(i_r) \leq \text{finish}(j_r)$

Inductive Hypothesis: $\text{finish}(i_{r-1}) \leq \text{finish}(j_{r-1})$

$\text{finish}(j_{r-1}) \leq \text{start}(j_r)$

$\text{finish}(i_{r-1}) \leq \text{start}(j_r)$

$\text{finish}(i_r) \leq \text{finish}(j_r)$

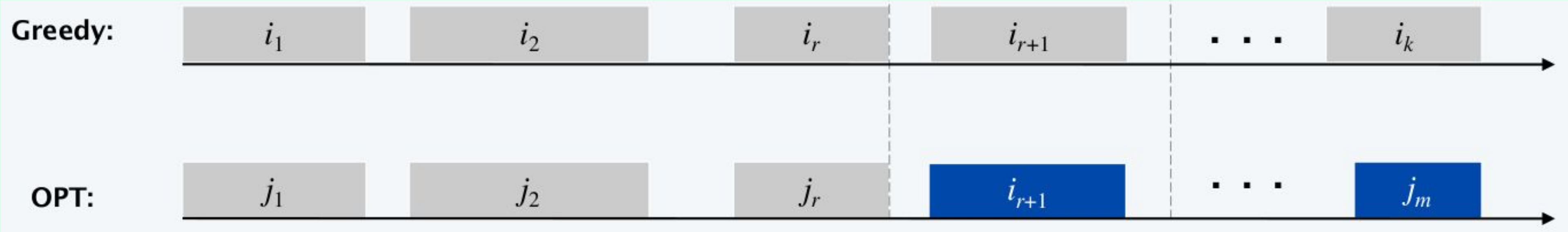


Interval Scheduling - Earliest-finish-time-first (EFTF)

Prove that the greedy algorithm returns an optimal set A

Proof by Contradiction: If A is not optimal then $m > k$

i.e. OPT has more intervals than A



Interval Scheduling - Earliest-finish-time-first (EFTF)

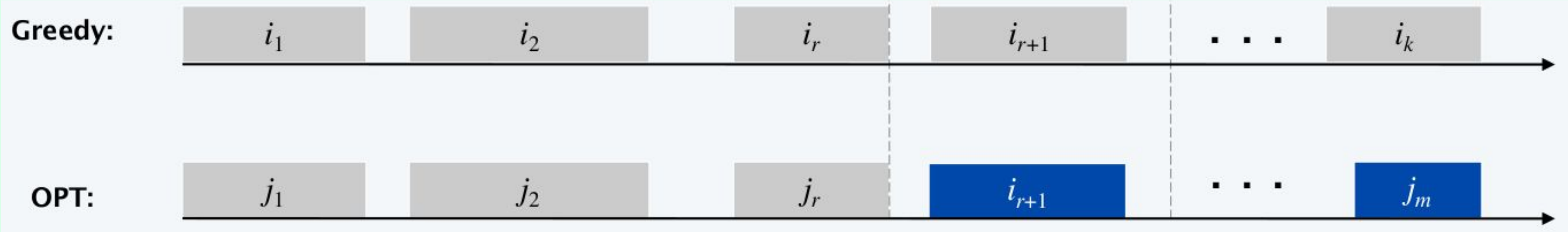
Prove that the greedy algorithm returns an optimal set A

<u>Proof by Contradiction:</u> If A is not optimal then $m > k$

i.e. OPT has more intervals than A

We know finish(i_k) $\leq$ finish(j_k)

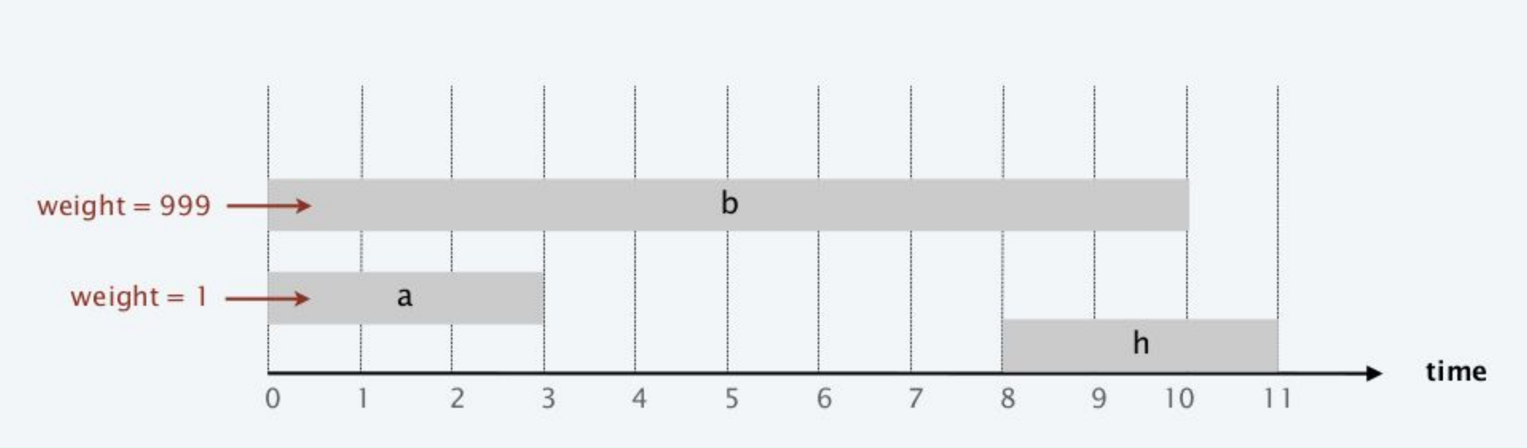
But OPT has interval j_{k+1} which is compatible with the intervals in A - which is a contradiction.



Weighted interval scheduling

Suppose, in addition to the start and finish times, we add a weight to each job and now you have to find the subset of jobs that maximize the overall weight. Can we still find the optimal solution through the EFTF algorithm?

Weighted interval scheduling



Counter-example

Reference reading:

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