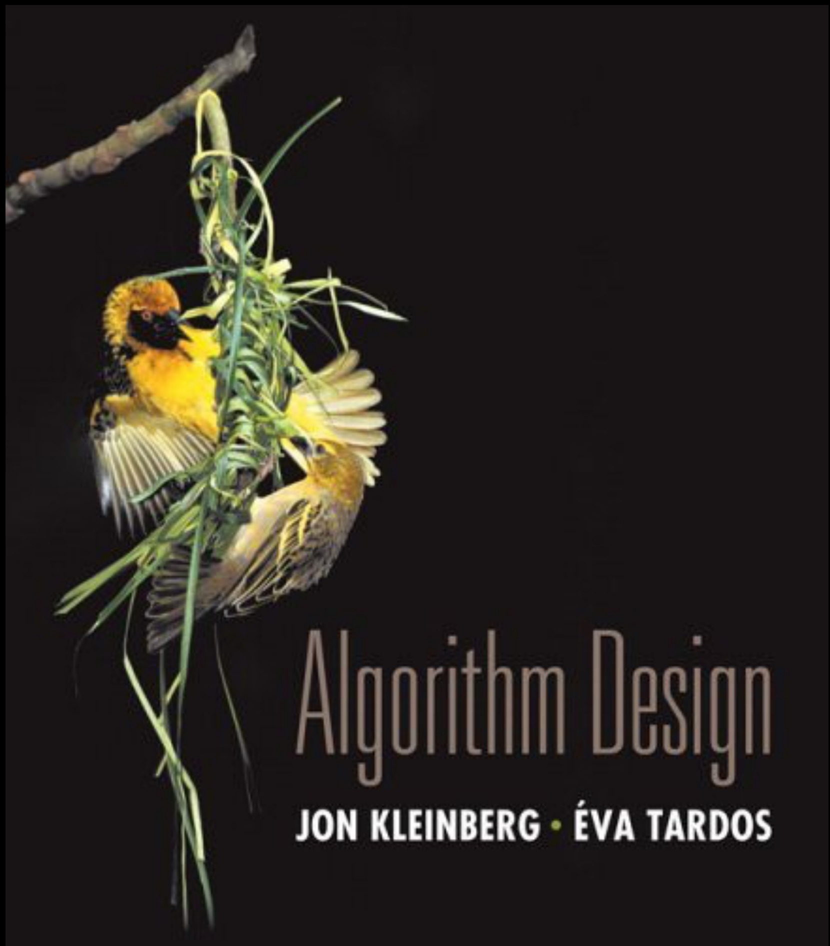


Chapter 4

Greedy Algorithms

Slides by Kevin Wayne.

4.1 Interval Scheduling

Interval Scheduling

EXERCISE I:

Prove that the Greedy Algorithm based on the **earliest finish time** is **optimal**

Hint: Use Lemma 1 (Greedy stays ahead)

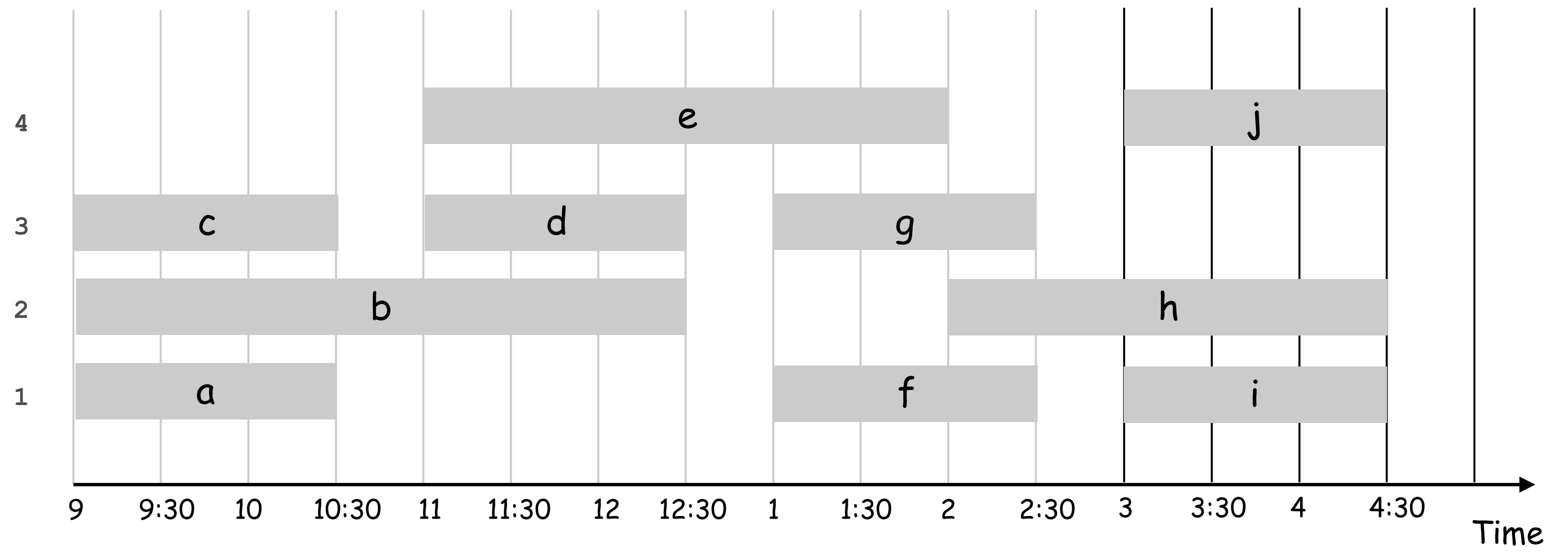
4.1 Interval Partitioning

Interval Partitioning

Interval partitioning.

- Lecture j starts at s_j and finishes at f_j.
- Goal: find **minimum** number of **classrooms** to schedule **all** lectures so that no two occur at the same time in the same room.

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

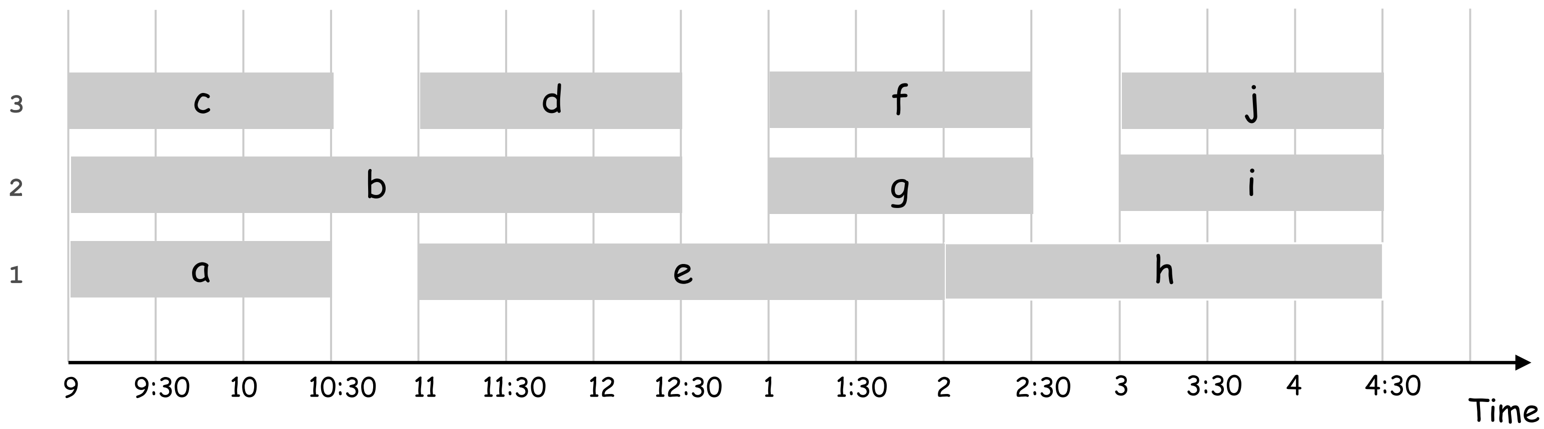


Interval Partitioning

Interval partitioning.

- Lecture j starts at s_j and finishes at f_j.
- Goal: find minimum number of classrooms to schedule all lectures so that no two occur at the same time in the same room.

Ex: This schedule uses only 3.



Interval Partitioning: Lower Bound on Optimal Solution

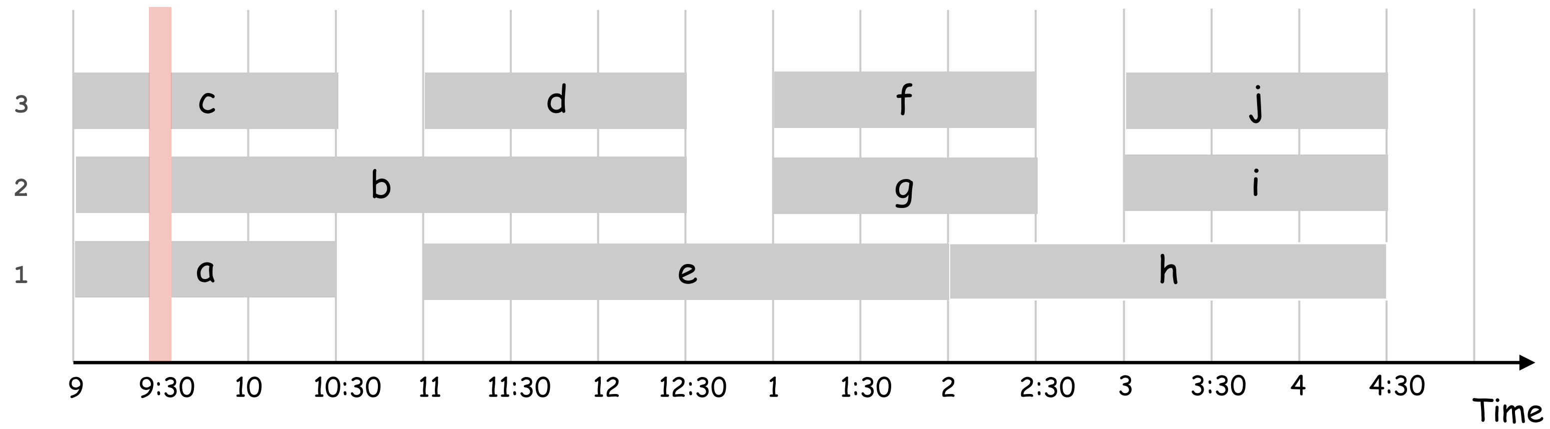
Def. The **depth** of a set of open intervals is the **maximum** number that contain any given time.

Key observation. **Number** of classrooms **needed** ≥ depth.

Ex: **depth** of schedule below = 3 ⇒ schedule below is **optimal**.

a, b, c all contain 9:30

Q. Does there always exist a schedule equal to **depth** of intervals?



Interval Partitioning: Greedy Algorithm

Greedy algorithm. Consider lectures in increasing order of **start time**: assign lecture to **any compatible**classroom.

```
Sort intervals by starting time so that s1 ≤ s2 ≤ ... ≤ sn.
d ← 0   ← number of allocated classrooms

for j = 1 to n {
   if (lecture j is compatible with some classroom k)
      schedule lecture j in classroom k
   else
      allocate a new classroom d + 1
      schedule lecture j in classroom d + 1
      d ← d + 1
}
```

Implementation. O(**n log n**).

- For each classroom k, maintain the finish time of the last job added.
- Keep the classrooms in a priority queue.

Interval Partitioning: Greedy Analysis

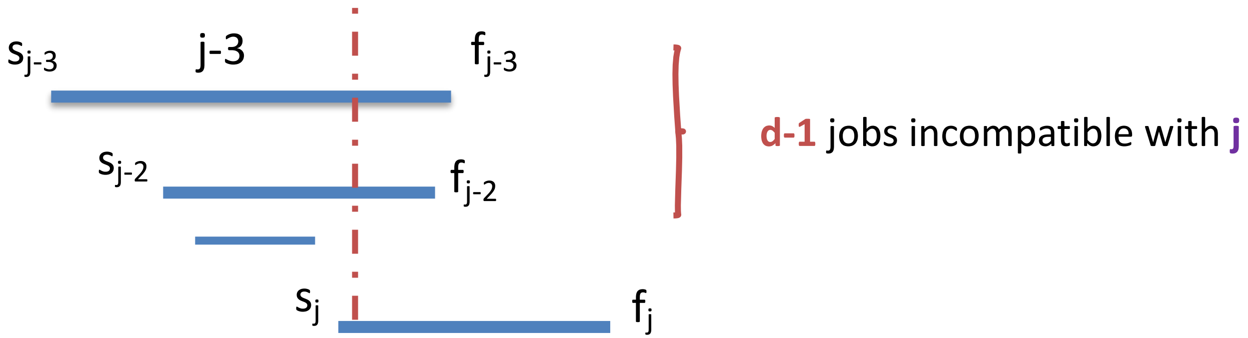
Observation. Greedy algorithm **never** schedules two incompatible lectures in the same classroom.

Theorem. Greedy algorithm is optimal.

Pf.

- Let **d** = number of classrooms that the greedy algorithm allocates.
- Classroom **d** is opened because we needed to schedule a job, say *j*, that is **incompatible** with all **d-1** other classrooms (i.e. d-1 jobs)
- These d jobs, **each must** end after s_j. ***(since they are incompatible with j)***
- Since we sorted by **start time**, all these incompatibilities are caused by **jobs** *that start* ***no later than*** s_j.
- Thus, we have **d** jobs overlapping at time $s_j + \varepsilon$.
- **Key observation (Lower Bound)** $\Rightarrow$ all schedules use $\geq$ **d** classrooms.

▪



EXCERCISE 1 – PAGE 183

- DON'T READ THE SOLUTION on THE BOOK !
- Follow this process:

1. Read the text and give a rigorous model for the instance and the goal
2. Describe a clear GREEDY APPROACH
3. Use «GREEDY STAYS AHEAD" to prove optimality in a way similar to Interval Scheduling
    1. Try to FORMALIZE A USEFUL LEMMA similar to LEMMA 1 there