Interval Scheduling

Correctness – Step-by-step approach

- A is a compatible set of requests
- For all indices \( r \leq k \) we have \( f(i_r) \leq f(j_r) \).
- The greedy algorithm returns an optimal set \( A \)

Running time

Interval Partitioning

Interval partitioning:
- Lecture \( j \) starts at \( a_j \) and finishes at \( b_j \).
- Goal: find minimum number of classrooms to schedule all lectures so that no two lectures occur at the same time in the same room.

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

Lower bound on optimal solution

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

Key observation. Number of classrooms needed = depth.

Q. Does minimum number of classrooms needed always equal depth?
A. Yes! Moreover, earliest-start-time first algorithm finds a schedule whose number of classrooms equals the depth.
Scheduling to minimize lateness

**Setting:**
- A single resource
- Each job \( j \) requires \( t_j \) time units, and a deadline of \( d_j \)
- \( j \) starts at \( s_j \), and is done at \( f_j = s_j + t_j \)
- The lateness of a job \( j \) is \( l_j = f_j - d_j \)

**Goal:**
- Decide on the start time of each job to minimize the maximum lateness (\( L \))
  \[ L = \max_j l_j \]

---

**Earliest deadline first**

**Correctness – Exchange approach**

**What is it?**
- We start with an optimal solution \( O \), and turn it into the greedy solution \( A \)

**For the earliest deadline first algorithm...**
- There exists an optimal schedule with no idle time
- The earliest-deadline-first schedule has no idle time
- The earliest-deadline-first schedule has no inversions
- All schedules with no inversions and no idle time have the same maximum lateness
- If a schedule (with no idle time) has an inversion, it has one with a pair of inverted jobs scheduled consecutively
- Swapping two adjacent, inverted jobs reduces the number of inversions by one and does not increase the max lateness

There is an optimal schedule that has no inversions and no idle time.

The schedule \( A \) produced by the greedy algorithm has optimal maximum lateness \( L \).