Interval Scheduling

Reading: Section 4.1 and 4.2
Problem 1

- Set of $n$ jobs to be executed
- Each job has a start time and a finish time: $s_1, \ldots, s_n, f_1, \ldots, f_n$
- Jobs cannot run in parallel
  - Jobs are compatible if their time does not overlap

**Goal:** Find maximum set of compatible jobs
Greedy Approach

- Simple rule to select first job
- Eliminate incompatible remaining jobs
- Repeat
Greedy Approach

- How should we pick the next job to schedule?
  - Earliest start time
  - Shortest interval
  - Fewest conflicts
  - Earliest finish time

- Exercise
Algorithm 1

Initially let $R$ be the set of all requests, and let $A$ be empty

While $R$ is not yet empty
  Choose a request $i \in R$ that has the smallest finishing time
  Add request $i$ to $A$
  Delete all requests from $R$ that are not compatible with request $i$
EndWhile

Return the set $A$ as the set of accepted requests

Running time: ??
Problem 2

- Set of $n$ jobs to be executed
- Each job has a deadline and an execution time: $d_1, \ldots, d_n, t_1, \ldots, t_n$
- Jobs cannot run in parallel
- All jobs must be executed

**Goal:** Find scheduling that minimizes maximum lateness
Greedy Approach

- How should we order jobs?
  - Increasing execution time
  - Slack time
  - Earliest deadline
Algorithm 2

Order the jobs in order of their deadlines
Assume for simplicity of notation that \( d_1 \leq \ldots \leq d_n \)
Initially, \( f = s \)
Consider the jobs \( i = 1, \ldots, n \) in this order
    Assign job \( i \) to the time interval from \( s(i) = f \) to \( f(i) = f + t_i \)
    Let \( f = f + t_i \)
End
Return the set of scheduled intervals \([s(i), f(i)]\) for \( i = 1, \ldots, n \)