Lecture 10 – Greedy algorithms

Reading: KT Section 4.1

Greedy algorithms

• An algorithm is greedy if it builds up a solution in small steps,
  • choosing a decision at each step myopically to optimize some underlying criterion.

• One can often design many different greedy algorithms for the same problem,
  • each one locally, incrementally optimizing some different measure on its way to a solution.

• When a greedy algorithm succeeds in solving a nontrivial problem optimally,
  • it typically implies something interesting and useful about the structure of the problem itself
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: greedy algorithms

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

Let's analyze!

**E**arliest-Finish-T**i**me-F**i**rst \( (n, s_1, s_2, \ldots, s_n, f_1, f_2, \ldots, f_n) \)

1. **Sort** jobs by finish time so that \( f_1 \leq f_2 \leq \ldots \leq f_n \)
2. \( A \leftarrow \emptyset \) **set of jobs selected**
3. **For** \( j = 1 \) **to** \( n \)
   - **If** job \( j \) is compatible with \( A \)
     - \( A \leftarrow A \cup \{ j \} \)
4. **Return** \( A \)