Lecture 9 – 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 are 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

**Earliest-Finish-Time-First** \((n, s_1, s_2, ..., s_n, f_1, f_2, ..., f_n)\)

**Sort** jobs by finish time so that \(f_1 \leq f_2 \leq ... \leq f_n\)

\(A \leftarrow \emptyset\) \hspace{2cm} \text{set of jobs selected}

**For** \(j = 1\) \(\text{to}\) \(n\)

**If** job \(j\) is compatible with \(A\)

\(A \leftarrow A \cup \{j\}\)

**Return** \(A\)