CS 7800: Advanced Algorithms

Class 2: Greedy Algorithms

Jonathan Ullman
September 9, 2025

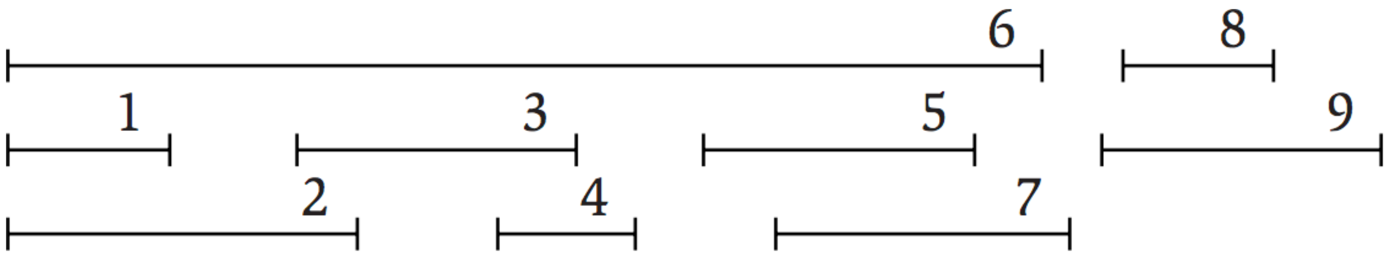
Optimization

Greedy Algorithms

- What's a greedy algorithm?
  - You know it when you see it
  - Typically builds a solution in one "pass" over the data
- Why care about greedy algorithms?
  - Fastest and simplest algorithms imaginable
  - Greedy algorithms are often useful heuristics
  - Greedy algorithms often arise naturally
  - Interesting proof techniques
    - Induction ("Greedy Stays Ahead")
    - Exchange Argument
    - Duality
    - ...

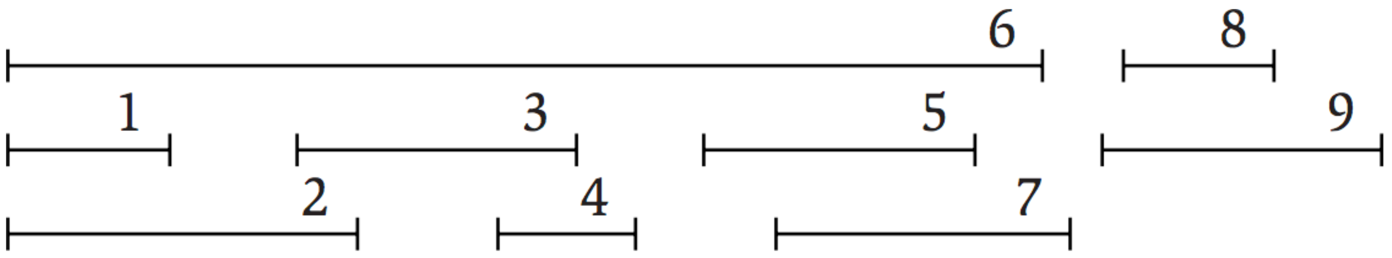
Interval Scheduling

- Input: n intervals (s_i, f_i)
- Output: a compatible schedule S with the largest possible size
  - A schedule is a subset of intervals $S \subseteq \{1, \dots, n\}$
  - A schedule S is compatible if no two $i, j \in S$ overlap
  - The size of the schedule is $|S|$



Generic Greedy Algorithm

- Sort intervals by [...]
- Let S be empty
- For $i = 1, \ldots, n$:
  - If interval i doesn't create a conflict, add i to S
- Return S



Possibly Correct Greedy Rules

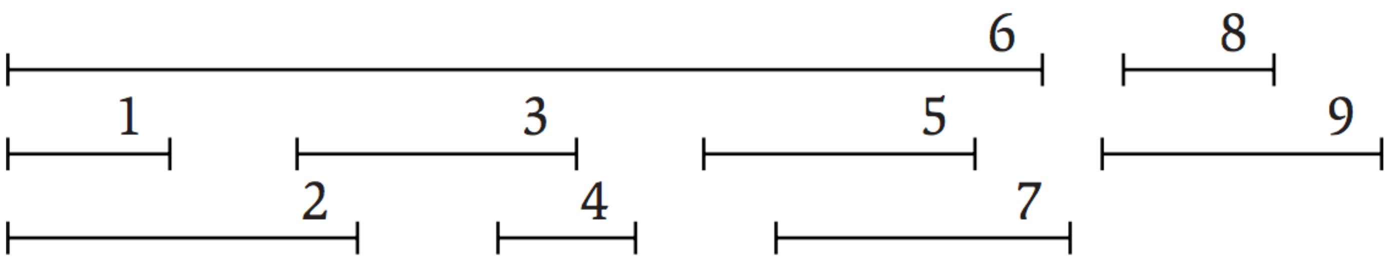
- Choose the shortest interval first

- Choose the interval with earliest start first

- Choose the interval with earliest finish first

Greedy Algorithm: Earliest Finish First

- Sort intervals so that $f_1 \le f_2 \le \cdots \le f_n$
- Let S be empty
- For $i = 1, \ldots, n$:
  - If interval i doesn't create a conflict, add i to S
- Return S



Greedy Stays Ahead

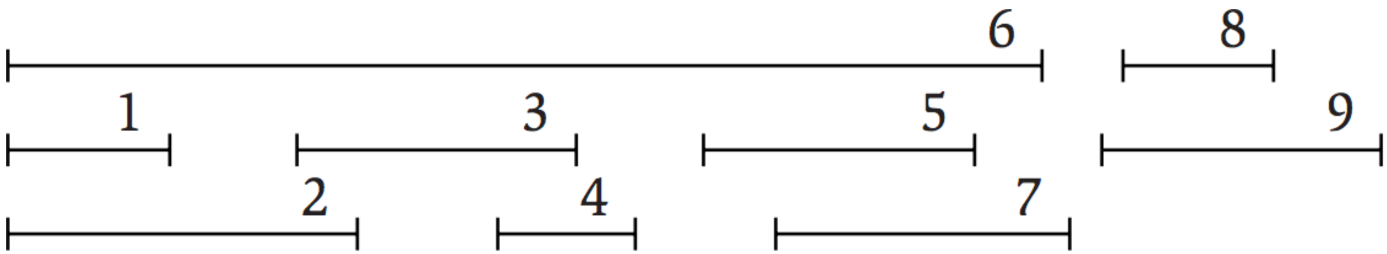
Greedy Stays Ahead

Greedy Stays Ahead

Greedy Stays Ahead

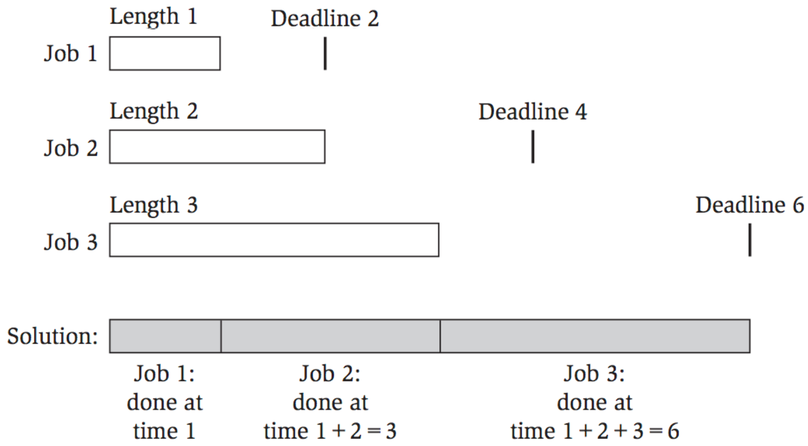
Interval Scheduling Recap

- There is an $O(n \log n)$ time greedy algorithm for the interval scheduling problem
  - Sort intervals by finish time, make one pass over the intervals, and add every compatible interval
  - Analyze using induction ("greedy stays ahead")



Minimum Lateness Scheduling

- Input: n jobs with length t_i and deadline d_i
  - Simplifying assumption: all deadlines are distinct
- Output: a minimum-lateness schedule for the jobs
  - Job i starts at s_i finishes f_i, no jobs overlap
  - The lateness of job i is $\max\{f_i - d_i, 0\}$
  - The lateness of a schedule is $\max_i\{\max\{f_i - d_i, 0\}\}$



Generic Greedy Algorithm

Possible Greedy Rules

Greedy Algorithm: Earliest Deadline First

- Sort jobs so that $d_1 \leq d_2 \leq \cdots \leq d_n$
- For $i = 1, \ldots, n$:
  - Schedule job i right after job $i - 1$ finishes

Exchange Argument

Exchange Argument

Exchange Argument

Exchange Argument

Exchange Argument

Exchange Argument

- **Putting the steps together (a thought experiment)**
  - (1) The greedy schedule G has no inversions
  - (2) While O is **not** equal to G
    - (2a) O has at least one inversion
    - (2b) O has a pair of consecutive jobs i, j that are inverted
    - (2c) Swap the order of i, j to fix the inversion
  - (3) Now O is equal to G but its lateness didn't increase, so O started at least as late as G

Minimum-Lateness Scheduling Recap

- There is an $O(n \log n)$ greedy algorithm for the minimum-lateness scheduling problem
  - Sort by earliest deadline and schedule jobs consecutively with no gaps
  - Analyze via an exchange argument