CSCI 3100: Algorithms

Greedy Algs (p# 2)
Announcements
Overall greedy strategy:

- Assume optimal is different than greedy
- Find the "first" place they differ.
- Argue that we can exchange the two without making optimal worse.

⇒ there is no "first place" where they must differ, so greedy in fact is an optimal solution.

Another example in notes: storing the most files on a tape

Intuition: (check notes)
A different scheduling problem:

Setting: a single resource (i.e., a processor/CPU)

Input: $n$ requests, each with:

- $D[i]$: deadline, $D[i]$ is the time when request $i$ should be completed
- $T[i]$: $T[i]$ is amount of time that request $i$ needs on the resource

Goal: Run everything.

If not everything can be finished by its deadline, then minimize the worst "lateness"
Lateness: Let's formalize.

If i gets finish time $F[i]$, $gF[i]$ is safe.

\[
\text{lateness } L[i] = F[i] - D[i]
\]

Goal:
Example: Job 1: □ | ▲
   length 1 | deadline 2

Job 2: □ □ | ▲
   length 2 | deadline 4
   length 3 | deadline 6

Input: D
T:

Schedule:

lateness?
Example: Job 1: \[
\begin{array}{c}
\text{length 1} \\
\text{deadline 2}
\end{array}
\]

Job 2: \[
\begin{array}{c}
\text{length 2} \\
\text{length 3} \\
\text{deadline 3} \\
\text{deadline 4}
\end{array}
\]

Schedule?
Ideas for how to be greedy:
Earliest deadline first (EDF)

Sort by \( D[i] \) + run in this order.

Sort of hard to believe this works!

That's why the proof is key.

But first: run time?
Correctness:

First: We may assume the optimal will have no idle time.

Why?

Schedule:
Definition: I'll say 2 jobs are inverted if job i goes before job j in the schedule, but $D[i] > D[j]$. 

Note: our greedy schedule has no inversions.

Lemma: All schedules with no inversions and no idle time have the same max lateness.

pf:
Thm: There is an optimal schedule with no inversions.

pf: Suppose opt must have inversions. Then $D[a] > D[b]$ but

opt: $\cdots a \cdots b$
pf cont.: So consider adjacent inversions, \( i \) and \( j \).

\[
\begin{align*}
\text{idea: } \quad & \quad \text{Swap them!} \\
& \quad \text{Know } j \text{ gets better.} \\
& \quad \text{What about } i? \\
\end{align*}
\]
Formalize: Worried about \( i \):

After swap, \( i \) finishes at \( F[j] \) from 1st schedule.

New lateness for \( i \):

\[ F[j] - D[i] \]

What was \( j \)'s before the swap?
Finally:

Since things only get better if we fix inversions, can just keep swapping. Will we reach a schedule with none?

ie - How many inversions can there be in the worst case?