Algorithms

Greedy algs
(pt 1)

Recap

- Practice questions:

  I'll hand out a practice exam.
  Also many on course website.

- HW3 - due Monday

- HW4 - up Monday, oral grading next Wednesday

  PLEASE SIGN UP!!

Dynamic Programing vs Greedy

Dyn. pro: try all possibilities
↳ but intelligently!

In greedy algorithms, we avoid building all possibilities.

How?
- Some part of the problem's structure lets us pick a local "best" and have it lead to a global best.

But - be careful!

Most common mistake:

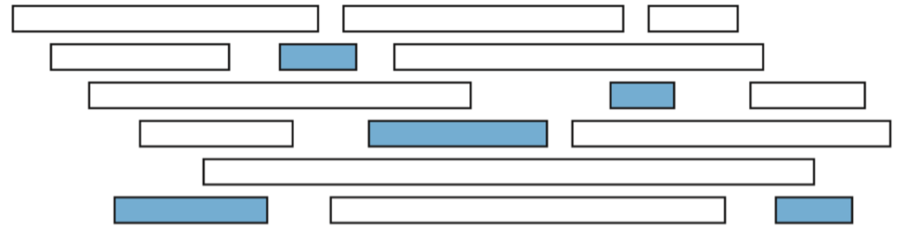
Students often design a greedy strategy, but don't check that it yields the best global one.

Example:

How to address?

Problem: Interval Scheduling

Given a set of events (intervals with a start + end time), select as many as possible so that no 2 chosen will overlap.



A maximal conflict-free schedule for a set of classes.

More formally

Input:

Output:

How would we formalize a dynamic programing approach?

Recursive structure:

Intuition for greedy:

Consider what might be a good first one to choose.

Ideas?

- Pick earliest start time?

- Latest end time?

- Shortest interval?

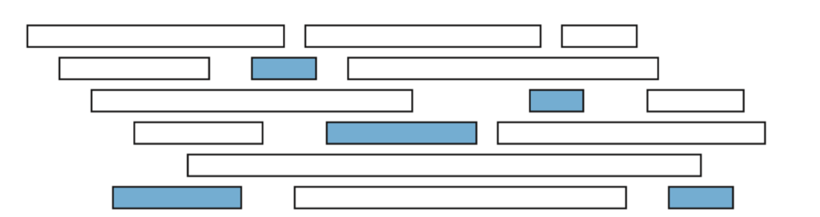
Instead:

Key intuition:

If it finishes as early as possible, we can fit more things in!

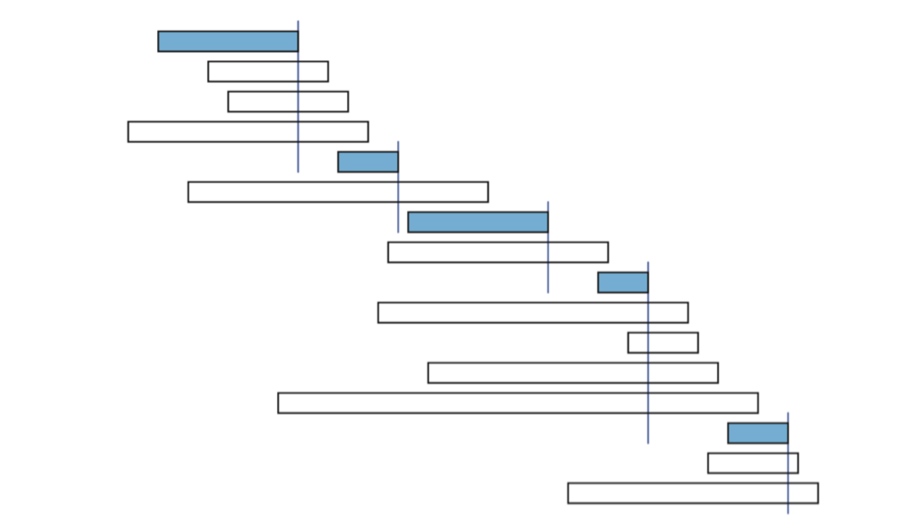
So - strategy:

Picture:



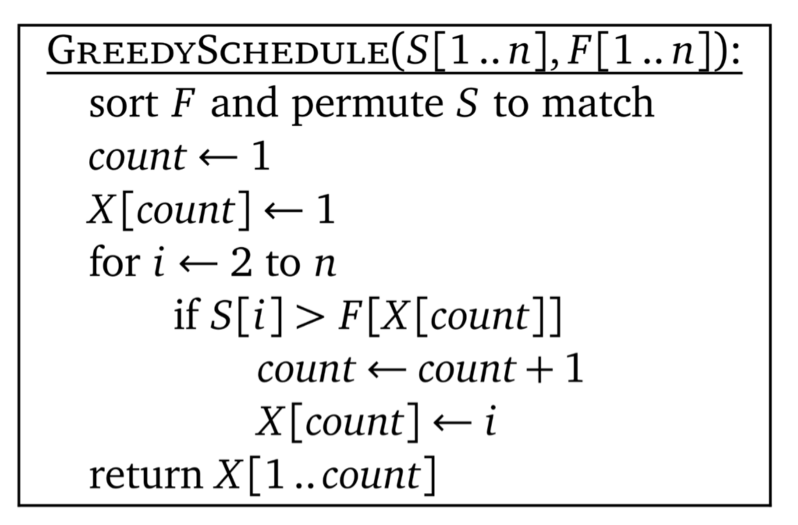
A maximal conflict-free schedule for a set of classes.

↓



The same classes sorted by finish times and the greedy schedule.

Pseudocode



```
GreedySchedule($S[1..n], F[1..n]$):
    sort $F$ and permute $S$ to match
    $count \leftarrow 1$
    $X[count] \leftarrow 1$
    for $i \leftarrow 2$ to $n$
        if $S[i] > F[X[count]]$
            $count \leftarrow count + 1$
            $X[count] \leftarrow i$
    return $X[1..count]$
```

Runtime :

Correctnes :

Why does this work?

Note: No longer trying all possibilities or relying on optimal substructure!

So we need to be *very* careful on our proofs.

(Clearly, intuition can be wrong!)

Lemma: We may assume the optimal schedule includes the class that finishes first.

pf: By contradiction:

Thm: The greedy schedule is optimal.

Pf: Suppose not.

Then $\exists$ an optimal schedule that has more intervals than the greedy one.

Consider first time they differ:

greedy:

$\langle g_1, g_2, g_3, \ldots, g_\ell \rangle$

optimal:

pf cont

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: