Scientific Computing

Jan 27, 2025

Announcements

→ Office Hours: Mondays + Fridays, 9:30 – 10:30
Cudahy 307

→ HW 1 assigned
On D2L → Dropbox
Due Friday, Jan 31

* Acceptable Sources: Online searches for how to do things in Python cite!

Unacceptable: Searching for the questions, AI Tools

Today

→ Greedy Algorithms

Problem #1: Interval Scheduling (Algorithm Design, by Kleinberg + Tardos)

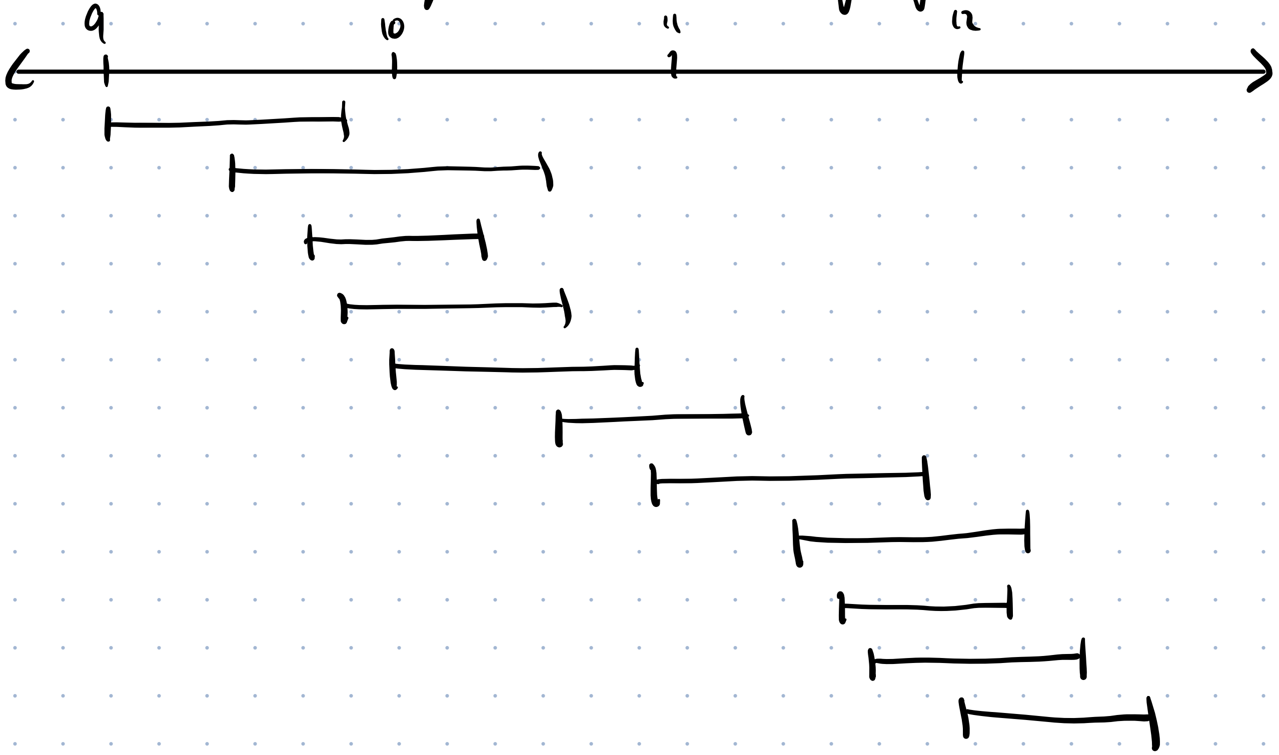
Suppose you are in charge of a conference room that a lot of people want to use to hold meetings. A bunch of people tell you the times they want to book the room for, and your goal is to accomodate as many groups as possible.

Ex:

Reservations:

9am - 9:50am
9:30am - 10:30am
9:45am - 10:15am
9:50am - 10:30am
10:00am - 10:50am
10:30am - 11:15am
11:00am - 11:50am
11:30am - 12:15pm
11:35am - 12:10pm
11:40am - 12:20pm
12:00pm - 12:30pm

What is the largest # of meetings you can book?



Formal setup: $[(s_1, f_1), (s_2, f_2), \ldots, (s_n, f_n)]$

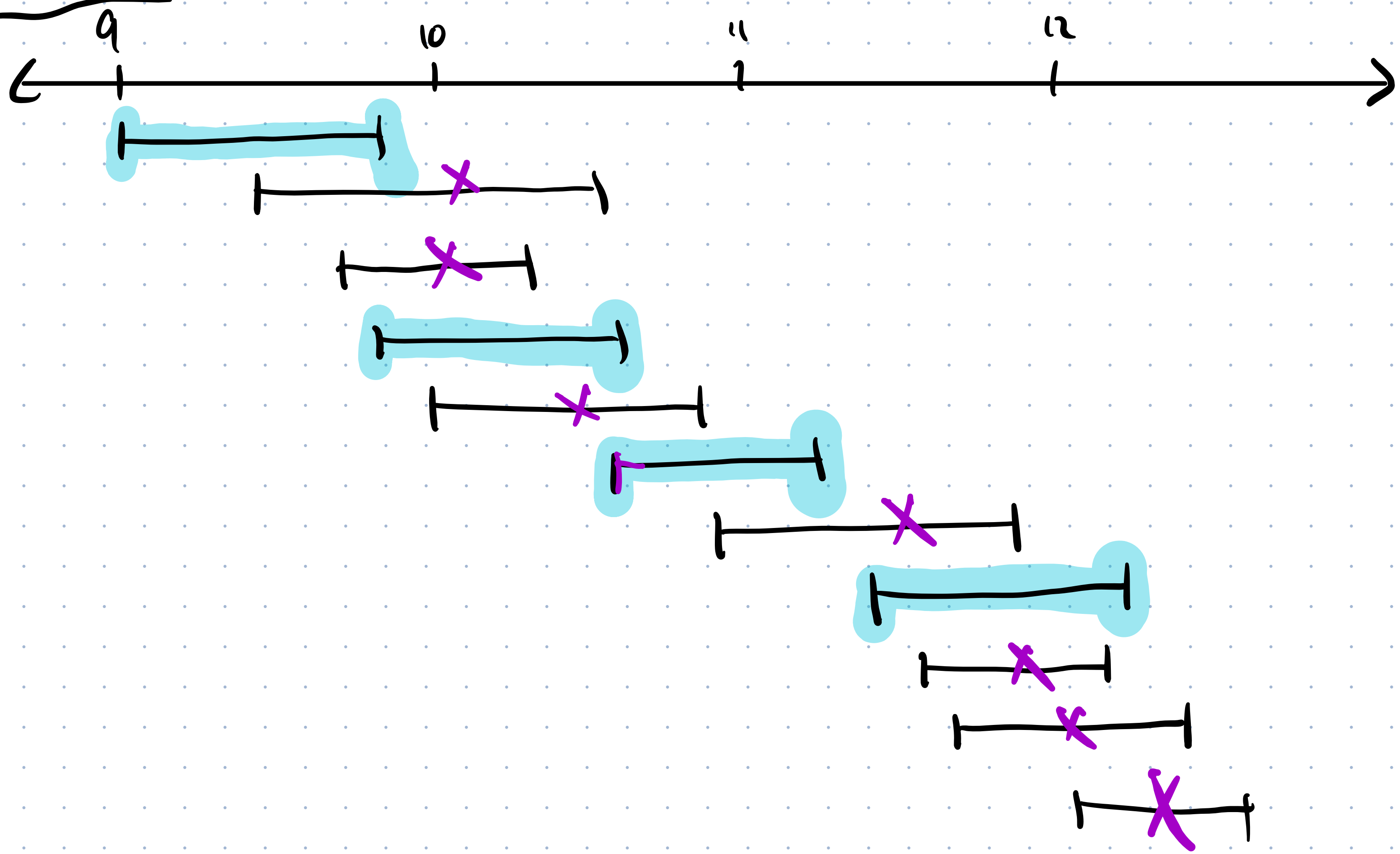
- n requests
- each request has a start time $\underline{s_i}$ and a finish time $\underline{f_i}$ (real numbers), with $s_i < f_i$.

Goal: find a maximal size subset of nonoverlapping requests

Two requests (s_i, f_i) and (s_j, f_j) overlap if:

$$\boxed{s_j < f_i \quad \text{and} \quad s_i < f_j}$$

Idea: best = earliest end time



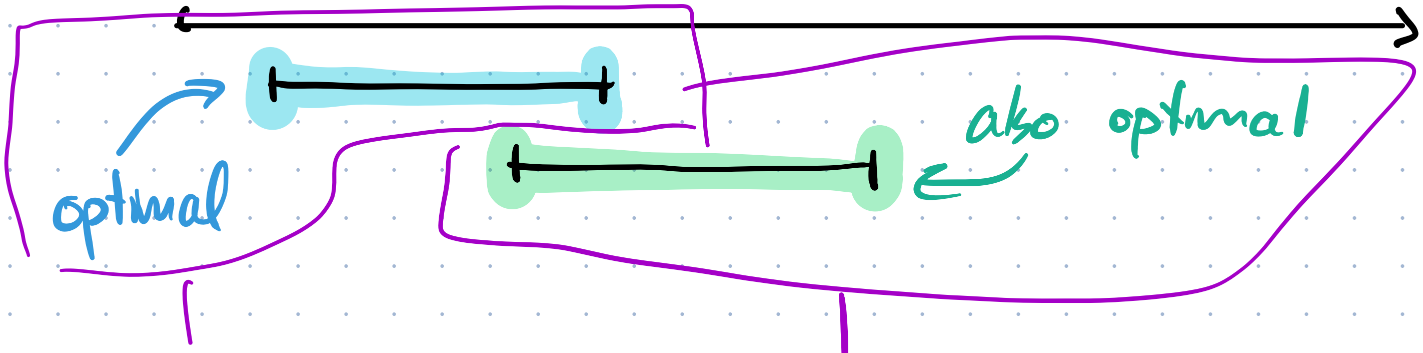
Algorithm:

```
Let R be the set of requests. $[(s_1,f_1),\ldots,(s_n,f_n)]$
Let A be the empty set. ← answer
While R is non-empty:
    Find the request with earliest end time.
                                   best
    Add it to A.
    Remove it from R and remove all other
      requests that are not compatible.
A is the solution
```

Theorem: The greedy algorithm always produces an optimal solution.

└ where best = earliest end time

Note: There could be other optimal solutions too.



greedy sol

diff sol, still optimal

- Coding the greedy algorithm!
    - Python lesson on functions and sort keys
    - Demo