Scientific Computing

Jan 31, 2025

Announcements

-> Office Hours: Mondays + Fridays, 9:30 - 10:30
Cudahy 307

-> HW 1 due tonight, 11:59pm

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

Unacceptable: Searching for the questions, AI Tools

Today

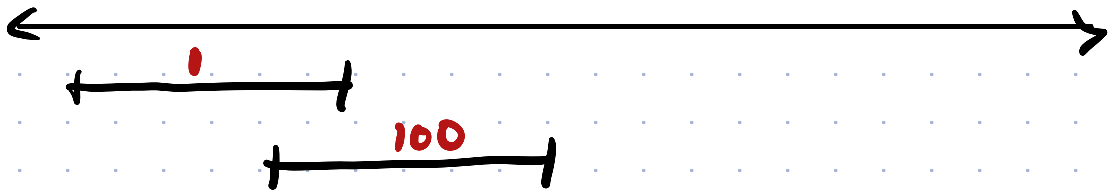
-> Greedy Algorithms

~~-> Unix command line (maybe)~~

Problem #3: Weighted Interval Scheduling

This is like regular interval scheduling, except each request i comes with a value v_i and your goal is to maximize the total value of satisfied requests.

Our previous greedy algorithm is now pretty bad.



Maximizing!

Possible Greedy Algos:

* Best = most profitable (highest weight)
* best = shortest
* best = least conflicting
* best = [of the 10 shortest meetings, the most profitable]
* best = highest $\frac{\text{profit}}{\text{duration}}$ # [if ties, go earliest end time]
  (profit/hour)

Are any of these optimal? No

[demo]

The is no known greedy algorithm that is optimal.

How long would brute force take? If there are n requests, you'd need to check all 2^n subsets of them. So, the run time would be exponential, something like $O(2^n)$ or $O(n \cdot 2^n)$.

This is "big-O" notation, and it tells you roughly how many steps an algorithm has to use.

For this particular problem, there is a technique to do it in $O(n \cdot \log(n))$ time – very fast!

"dynamic programming"