

Interval Scheduling

You have a single processor, and a set of jobs with fixed start and end times.

Your goal is to maximize the number of jobs you can process.

I.e. choose the maximum number of non-overlapping intervals.



Greedy Algorithm

Earliest end time

Latest end time

Earliest start time

Latest start time

Shortest interval

Fewest overlaps (with remaining intervals)

Summary

Greedy algorithms

You'll probably have 2 (or 3...or 6) ideas for greedy algorithms. Check some simple examples before you implement!

Greedy algorithms rarely work.

When they work AND you can prove they work, they're great!

Proofs are often tricky

Structural results are the hardest to come up with, but the most versatile.

Greedy stays ahead usually use induction

Exchange start with the **first** difference between greedy and optimal.

What Comes next?

CSE 417 is intended as a "last" undergraduate course for those of you about to graduate. But if you are hoping for more, consider:

CSE 422 (toolkit for modern algorithms)

Mathematics and algorithms for larger data sets/ML, approximate solutions

CSE 521 (Graduate algorithms)

More math background expected, more approximation algorithms, and a few other topics.

CSE 431 (Complexity theory)

Proof-based (all your homework problems are proofs). More on NP, undecidability, other notions of "hard."

Courses in Math & AMATH (especially for LPs and network flow)

Math 461, 462 (combinatorics); Math 407, 408 (LPs, convex optimization); Math 409 (lots of overlap with this course)

Math 514 (deep theory behind LPs, MSTs, etc.)