Learning Goals

- Describe typical properties of greedy algorithms + key words: "Objective Function", "Completion Time", "Optimization Problem", "Greedy Ordering," "Optimal ordering", "Scoring Function"
- Create counterexample for a greedy ordering algorithm [G1]

Greedy Algorithm (informal def): an alg that sequentially contstructs a solution through a series of myopic (short-sighted = local = not global = not thinking about future) decisions

Typical properties

- Easy to create
- Runtime easy to analyze
- Frequently not optimal
- When optimal, hard to prove correctness

Scheduling Tasks

"optimization problem"

task	1	2	3
time	3	4	2
weight	5	1	2

$t_1 = 3 \quad w_3 = 2$

Input: n tasks: time, weight for each

↑ larger = more important

Output: Order σ (sigma) that minimizes

"objective function" → $$A(\sigma) = \sum_{i=1}^{n} w_i C_i(\sigma)$$

$$\underbrace{\sigma \in \{(123), (213), \ldots}_{3!}$$

Applications: CPU scheduling, HW schedules

Brute Force $\Omega(n!)$

$C_i(\sigma)$ = completion time of task i w/ ordering σ

ex:

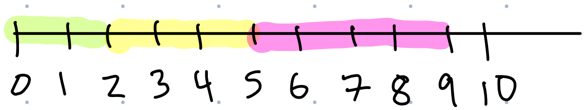
task	1	2	3
time	3	4	2

$\sigma = (3,1,2)$

$C_1(3,1,2) = 5$

$C_2(3,1,2) = 9$

$C_3(3,1,2) = 2$



What is $C_3(2, 1, 3)$?

A) 2 B) 3 C) 7 D) 9

Calculate $A(\sigma)$ + determine best order: (Brute force)

task	1	2
time	3	4
weight	5	1

Order	(1,2)	(2,1)
$A(\sigma)$	22	39

Order	(1,2)	(2,1)
$C_1(\sigma)$	3	7
$C_2(\sigma)$	7	4

$$A(\sigma) = \sum_{i=1}^{n} w_i C_i(\sigma)$$

$5 \cdot 3 + 1 \cdot 7 = 22$

$1 \cdot 4 + 5 \cdot 7 = 39$

What is this objective function optimizing? What time/weight jobs is it prioritizing?

<u>Output:</u> Ordering σ that minimizes

$$A(\sigma) = \sum_{i=1}^{n} w_i C_i(\sigma)$$

Creating Scoring Functions + Counterexamples

Creating a greedy algorithm

To create, come up with any function $f(i)$ of i, w_i, t_i

$$f(i) = t_i - 6 \quad \text{(order in increasing order)}$$

↑ scoring function

task	1	2
time	3	4
weight	5	1
f	-3	-2

We did brute force above, found (1, 2) is best

↑ Optimal Ordering

⇒ Greedy Ordering = (1, 2)

But will this greedy ordering always be correct?

[G1]

Try to find a counterexample:

- Think extreme!
- Create tasks close in score but different otherwise

$f(i) = t_i - 6$ (increasing)

task	1	2
time	1	2
weight	1	100
f	-5	-4

→ Greedy Ordering (1,2)

Order	(1,2)	(2,1)
$A(\sigma)$	301	203

Opt. Ordering (2,1)

Counterexample

Group Exercise [G1]

Create counterexamples for scoring functions

A) $f(i) = w_i$ (decreasing) B) $f(i) = w_i - t_i$ (decreasing)

A)

task	1	2
time	1000	1
weight	2	1
f	2	1

→ (1, 2)

Brute force:

Order	(1,2)	(2,1)
$A(\sigma)$	3001	2003

Optimal: (2, 1)

B)

task	1	2
time		
weight		
f		

Brute force

Order	(1,2)	(2,1)
$A(\sigma)$		

Optimal:

One approach to designing greedy alg:

- create several reasonable scoring functions
- test, try to create counterexamples
- if no counter example, try to prove correct, or just use (heuristic)

<u>Thm</u>: Ordering jobs by $f = \frac{w_i}{t_i}$ (decreasing) is optimal for minimizing $A(\sigma) = \sum_i w_i C_i(\sigma)$ if w_i/t_i are all distinct.

<u>Pf</u>: [Exchange Argument = Type of Pf by Contradiction]

WLOG, relabel so $w_1/t_1 > w_2/t_2 > w_3/t_3$ so greedy ordering is $\sigma = (1, 2, 3, \dots, n)$. Assume for contradiction that σ is not optimal. Then $\exists\, \sigma^*$ that is optimal,