

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 [G1]

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

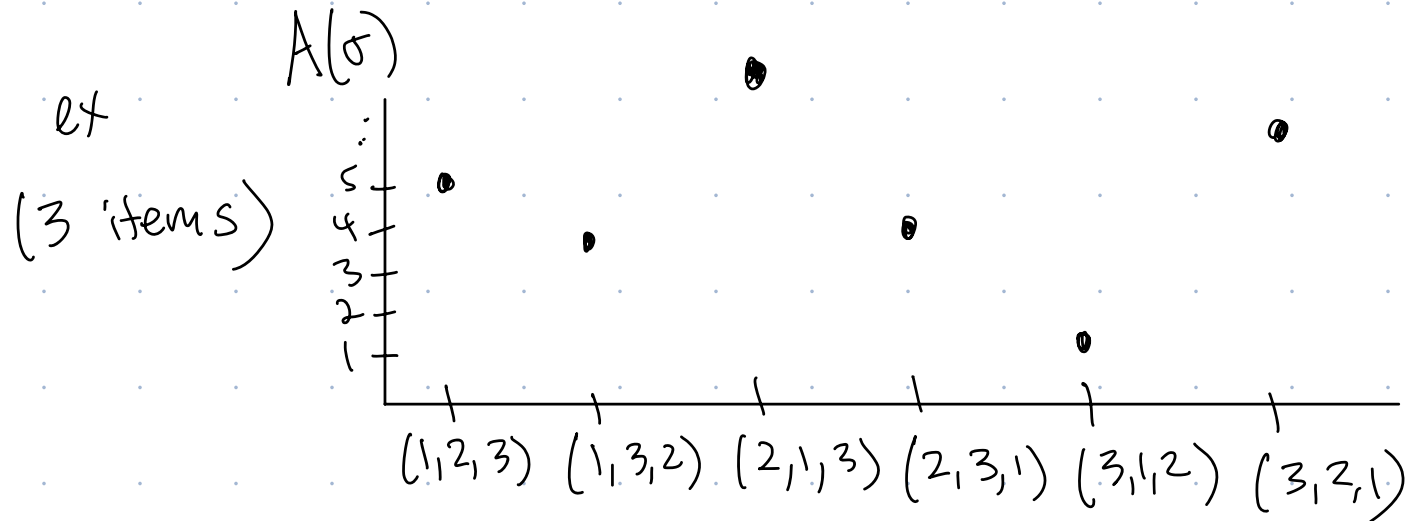
Typical properties

-
-
-
-

Ordering Problem (Abstract)

Input:

Output:



Brute Force: Evaluate $A(\sigma)$ for every ordering
+ find min-value ordering.

Greedy: • Design simple scoring function

• Evaluate score of each item

• Return σ that puts items in decreasing (increasing)
order.

Scheduling Tasks

Input:

task	1	2	3
time			
weight			

Output:

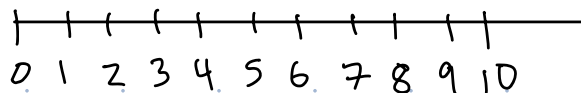
Applications:

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

ex:

task	1	2	3
time	3	4	2

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



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)
$A(\sigma)$		

Order	(1,2)	(2,1)
$C_1(\sigma)$		
$C_2(\sigma)$		

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

Output: 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 .

task	1	2
time	3	4
weight	5	1
f		

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

But will this greedy ordering always be correct?

Try to find a counterexample:

-
-

$$f(i) = t_i - b(\text{increasing})$$

task	1	2
time		
weight		
f		

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

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		
weight		
f		

Brute force:

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

Optimal:

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)

Thm: 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.

Pf: [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.