Discrete Optimization

The Knapsack Problem: Greedy Algorithms

Goals of the Lecture

- Compare different greedy algorithms
  - knapsack example

What is Greedy?

- Assume the it's "easy" to build a feasible solution

- Key Idea:
  - Build a solution by picking items one at a time

- Called: greedy algorithms or heuristics

The Temple is Collapsing

$1 Million
2kg

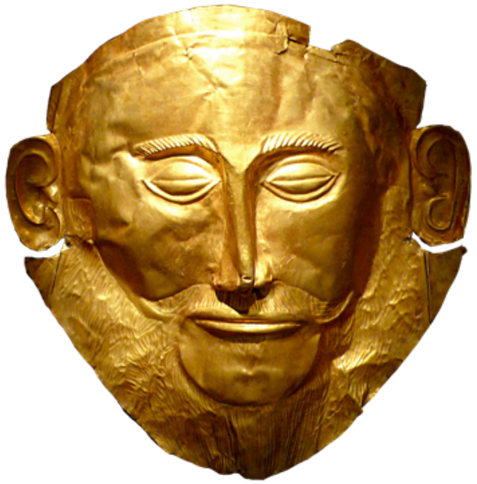
$1 Million
2kg

$1 Million
2kg

$10 Million
5kg

$10 Million
5kg

Which items to take?



$13 Million
8kg

$7 Million
3kg

Maximum
Capacity
10kg

Knapsack Greedy Algorithms

- Idea 1:
  - More items is best, start with small ones and take as many as you can

Idea: The More Items the Better

$1 Million
2kg

$1 Million
2kg

$1 Million
2kg

$10 Million
5kg

$10 Million
5kg

Total: $10 Million

$13 Million
8kg

$7 Million
3kg

Maximum
Capacity
10kg

Knapsack Greedy Algorithms

- Idea 1: ($10 Million)
  - More items is best, start with small ones and take as many as you can
- Idea 2:
  - Valuable items are best, start with the most valuable items

Idea: More Valuable is Better

$1 Million
2kg

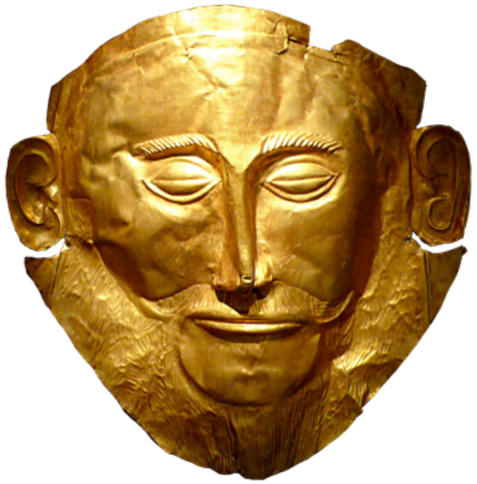
$1 Million
2kg

$1 Million
2kg

$10 Million
5kg

$10 Million
5kg

Total: $14 Million



$13 Million
8kg

$7 Million
3kg

Maximum
Capacity
10kg

Knapsack Greedy Algorithms

- Idea 1: ($10 Million)
  - More items is best, start with small ones and take as many as you can
- Idea 2: ($14 Million)
  - Valuable items are best, start with the most valuable items
- Idea 3:
  - Value density! dollars per kilogram

Idea: The More Items the Better

0.5
M/kg

$1 Million
2kg

$1 Million
2kg

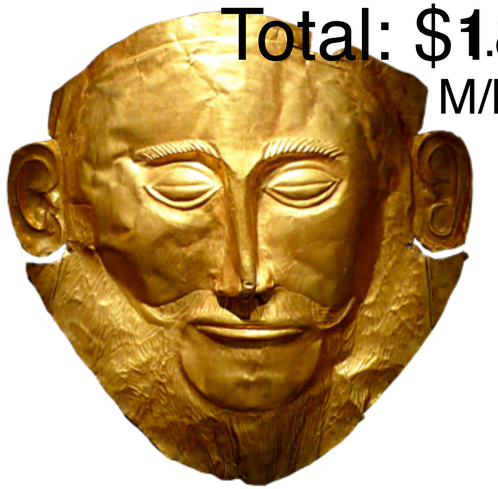
$1 Million
2kg

2.0
M/kg

$10 Million
5kg

$10 Million
5kg

Total: $18 Million



1.8
M/kg

$13 Million
8kg

2.3
M/kg

$7 Million
3kg

Maximum
Capacity
10kg

Knapsack Greedy Algorithms

- Is $18 million dollars the best we can do?
  - optimal?

Total: $20 Million!

$10 Million
5kg

$10 Million
5kg

Maximum
Capacity
10kg

Greedy Algorithms Overview

- For one problem, there are **many** possible greedy algorithms.
  - some will do better than others
  - depends on the input!
- Advantages
  - quick to design and implement
  - can be very fast
- Problems
  - no quality guarantees (in general)
  - quality can vary widely on the input
  - problem feasibility needs to be “easy”

The Essence of this Class

- We can always start with greedy

- Going beyond greedy
  - Constraint Programming
  - Local Search
  - Mixed Integer Programming

- Ways to
  - reliably find feasible solutions
  - reliably build high-quality solutions
    - robust to different inputs
  - ideally, proving those solutions are the best

Until Next Time

Citations

Stone Foundation Tablet with Inscription of Gudea - 41221 (http://commons.wikimedia.org/wiki/File:Sumerian _-_Stone_Foundation_Tablet_with_Inscription_of_Gudea_-_Walters_41221_-_View_A.jpg). Artist Unknown. Walters Art Museum [Public domain, CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

Stone Foundation Tablet with Inscription of Gudea - 41220 (http://commons.wikimedia.org/wiki/File:Sumerian_-_Stone_Foundation_Tablet_with_an_Inscription_of_Gudea_-_Walters_41220_-_View_A.jpg). Artist Unknown. Walters Art Museum [Public domain, CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0) via Wikimedia Commons

Ring with the engraved portrait of Ptolemy VI Philometor (http://commons.wikimedia.org/wiki/File:Ring_with_engraved_portrait_of_ Ptolemy_VI_Philometor_(3rd%E2%80%932nd_century_BCE)_-_2009.jpg<http://commons.wikimedia.org/wiki/File:Ring_with_engraved_portrait_of_Ptolemy_VI_Philometor_(3rd-2nd_century_BCE)_-_2009.jpg>) By Unknown. (Photographed by PHGCOM in 2009.) [Public domain], via Wikimedia Commons

the mask of agamemnon (http://www.flickr.com/photos/rosemania/5705122218/) by Xuan Che (http://www.flickr.com/people/rosemania/) CC BY-2.0 (http://creativecommons.org/licenses/by/2.0/deed.en)

Terracotta Warrior (http://www.flickr.com/photos/59627558@N00/4677378806/) by fixermark (http://www.flickr.com/photos/59627558@N00/) CC BY 2.0 (http://creativecommons.org/licenses/by/2.0/deed.en)