Design and Analysis of Algorithms

Fractional Knapsack

Instructor: Dinesh Khandelwal

The Knapsack Problem

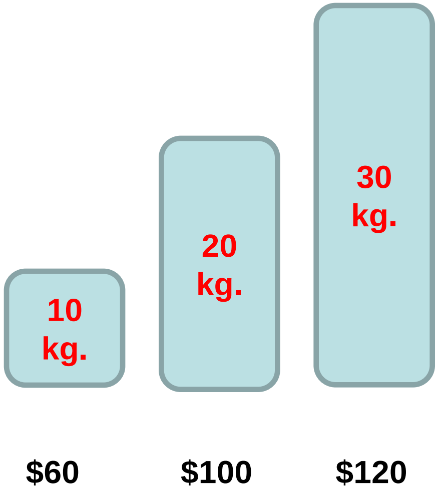
- The famous *knapsack problem*:
  - A thief breaks into a museum. Fabulous paintings, sculptures, and jewels are everywhere. The thief has a good eye for the value of these objects, and knows that each will fetch hundreds or thousands of dollars on the clandestine art collector's market. But, the thief has only brought a single knapsack to the scene of the robbery, and can take away only what he can carry. What items should the thief take to maximize the haul?

Fractional knapsack and 0-1 knapsack

- After you break into a house, how to choose the items you put into your knapsack, to maximize your income.
- Each item has a weight and a value
- Your knapsack has a maximum weight
- 0-1 knapsack
  - You can only take an item of leave it there
- Fractional knapsack
  - You can take part of an item

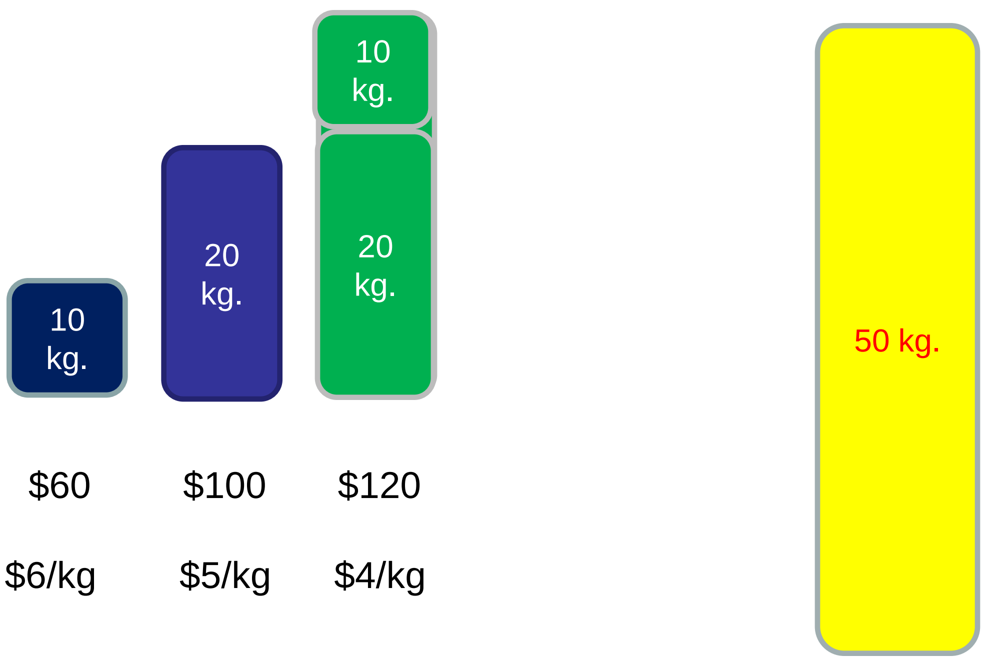
Fractional knapsack

- The **fractional knapsack problem** is a classic problem that can be solved by greedy algorithms
- E.g.
  - your knapsack can contain 50 Kg. Stuff;
  - the items are as in the figure
  - What is your algorithm?



Fractional knapsack

- A greedy algorithm for **fractional knapsack problem**
- Greedy choice: choose the maximum value/kg. item



Fractional Knapsack

Algorithm ***fractionalKnapsack***(S, W)

Input: set S of items ; benefit b_i and weight w_i; max. weight W

Output: amount x_i of each item I to maximize benefit with weight at most W

for ***each item i in S***

$x_i \leftarrow 0$

$v_i \leftarrow b_i / w_i$ {value}

$w \leftarrow 0$ {current total weight}

sort the array v_i

while $w < W$

remove item i with highest v_i

$x_i \leftarrow \min\{w_i, W - w\}$

$w \leftarrow w + \min\{w_i, W - w\}$

Fractional Knapsack

Time analysis:

O($n \log n$)	to sort the items by value
O(n)	to select amount of each item
O($n \log n$)	total

Fractional Knapsack

Claim. Greedy algorithm gives best optimal item set.

Proof: Sort the items by value per kg.

Let $OPT = \{y_1, y_2, \ldots, y_n\}$ be an optimal solution to fractional knapsack, consisting of amounts y_i for item $i = 1..n$.

We will transform OPT into the greedy solution $\{x_i\}_{1..n}$ using induction while never reducing the value of OPT.

Hence greedy must have the same value as OPT and it is also optimal.

Fractional Knapsack

Proof: OPT and greedy are sorted by value $\$ / \mathrm{kg}$

Basis of induction: $i = 1$. Clearly $y_1 \le x_1$ since the greedy choice loads as much weight as possible from the most valuable item ($\mathrm{price/weight}$).

If $y_1 = x_1$, then OPT and greedy match in the first selection, and there is nothing to show.

If $y_1 < x_1$ then remove $|x_1 - y_1|$ kgs from the least valuable items starting from y_m and replace them with the same amount of kgs of y_1.
Value of solution cannot go down as y_1 gives the best value per kg.

Fractional Knapsack

Proof:

Step of induction: Assume $y_l = x_l$ for $l < i$.
Observe that $y_i \leq x_i$ since the greedy strategy loads as much weight as possible from item i, given the choices up to l.

If $y_i = x_i$, then OPT and greedy match in their selection, and there is nothing to show.

If $y_i < x_i$ then remove $|x_i - y_i|$ kgs from the least valuable items starting from y_m and replace them with the same amount of y_i.
Value of solution cannot go down as y_i gives the best value per kg of y_k, $k = i..m$.

Fractional Knapsack

Proof:

Hence the value of OPT and greedy is the same
➔ greedy is optimal too.

0-1 knapsack

- The **0-1 knapsack problem** is a classic problem that can **not** be solved by greedy algorithms
- Can you design an algorithm of this problem?