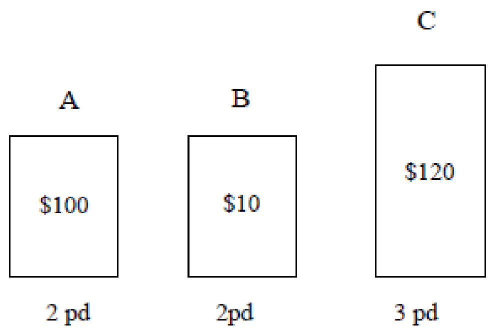
Greedy Algorithms: The Fractional Knapsack

Dr. Krishna Gopal Dhal
Assistant Professor
Dept. of Computer Science and Application
Midnapore College (Autonomous)

Introduction to Greedy Algorithm

- A greedy algorithm for an optimization problem always makes the choice that looks best at the moment and adds it to the current subsolution.
- Final output is an optimal solution.
- Greedy algorithms don't always yield optimal solutions but, when they do, they're usually the simplest and most efficient algorithms available.

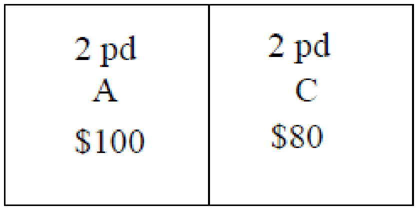
The Knapsack Problem...



Capacity of knapsack: $K = 4$

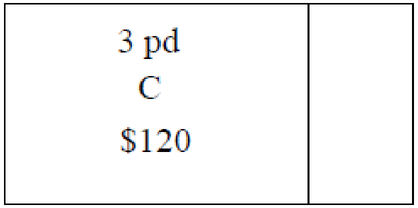
Fractional Knapsack Problem:
Can take a fraction of an item.

Solution:



0-1 Knapsack Problem:
Can only take or leave item. You can't take a fraction.

Solution:



The Fractional Knapsack Problem: Formal Definition

- Given K and a set of n items:

weight	w_1	w_2	$\ldots$	w_n
value	v_1	v_2	$\ldots$	v_n

- Find: $0 \leq x_i \leq 1$, $i = 1, 2, \ldots, n$ such that

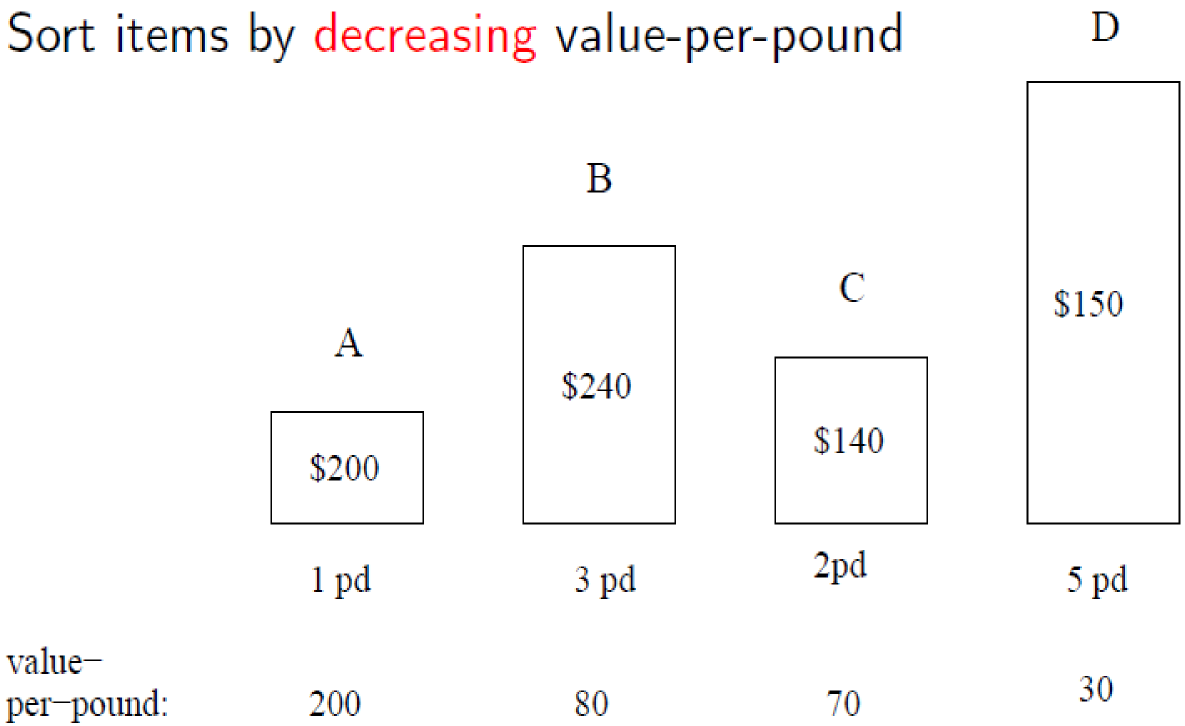
$$\sum_{i=1}^{n} x_i w_i \leq K$$

and the following is maximized:

$$\sum_{i=1}^{n} x_i v_i$$

Greedy Solution for Fractional Knapsack

Sort items by decreasing value-per-pound



If knapsack holds $K = 5$ pd, solution is:

1	pd	A
3	pd	B
1	pd	C

Greedy Solution for Fractional Knapsack

- Calculate the value-per-pound $\rho_i = \frac{v_i}{w_i}$ for $i = 1, 2, \ldots, n$.
- Sort the items by decreasing ρ_i.
  Let the sorted item sequence be $1, 2, \ldots, i, \ldots n$, and the corresponding value-per-pound and weight be ρ_i and w_i respectively.
- Let k be the current weight limit (Initially, $k = K$).
  In each iteration, we choose item i from the head of the unselected list.
  - If $k \geq w_i$, set $x_i = 1$ (we take item i), and reduce $k = k - w_i$, then consider the next unselected item.
  - If $k < w_i$, set $x_i = k/w_i$ (we take a fraction k/w_i of item i), Then the algorithm terminates.

Running time: $O(n \log n)$.

Greedy Solution for Fractional Knapsack

- Observe that the algorithm may take a fraction of an item. This can only be the last selected item.
- We claim that the total value for this set of items is the optimal value.