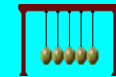
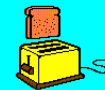


# Dynamic Programming

# 0/1 Knapsack Problem



# 0/1 Knapsack Problem



- Item  $i$  has a value (or profit)  $p_i$ .
- All weights and values are positive numbers.
- Hiker wants to select a subset of the  $n$  items such that
  - The weight of the subset should not exceed the capacity of the knapsack. (constraint)
  - Cannot select a fraction of an item. (constraint)
  - The value of the selected subset should be maximum. (optimization criterion)

# 0/1 Knapsack Problem



Let  $x_i = 1$  when item  $i$  is selected and let  $x_i = 0$  when item  $i$  is not selected.

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^n p_i x_i \\ &\text{subject to} && \sum_{i=1}^n w_i x_i \leq c \\ &&& \text{and } x_i = 0 \text{ or } 1 \text{ for all } i \end{aligned}$$

# Greedy Method 1

Be greedy on capacity utilization.

- Select items in increasing order of weight.

$$n = 2, c = 7$$

$$w = [3, 6]$$

$$p = [2, 10]$$

only item 1 is selected

the value of selection is 2

not best selection!

# Greedy Method 2

Be greedy on the value earned.

- Select items in the decreasing order of values

$$n = 3, c = 7$$

$$w = [7, 3, 2]$$

$$p = [10, 8, 6]$$

only item 1 is selected

the value of selection is 10

not best selection!

# Greedy Method 3

Be greedy on value density ( $p/w$ ).

- Select items in decreasing order of value density.

$$n = 2, c = 7$$

$$w = [1, 7]$$

$$p = [10, 20]$$

only item 1 is selected

the value of selection is 10

not best selection!