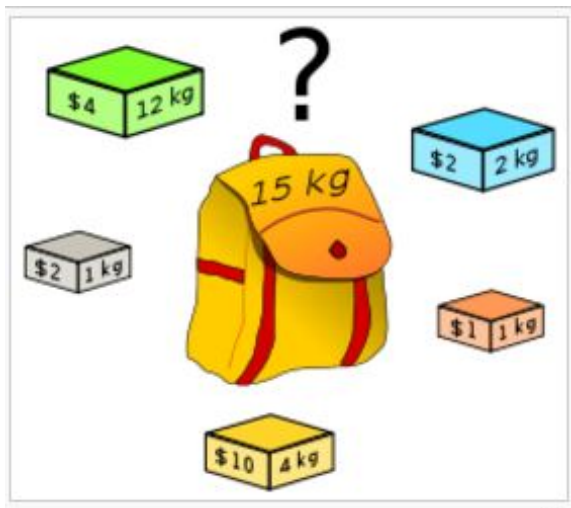# The Knapsack Problem

The Knapsack Problem

- Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

- It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items.

Knapsack Problems

- The knapsack problem or rucksack problem is a problem in combinatorial optimization.
- The problem often arises in resource allocation where there are financial constraints and is studied in fields such as combinatorics, computer science, complexity theory, cryptography and applied mathematics.

Knapsack Problems

- The knapsack problem has been studied for more than a century, with early works dating as far back as 1897.

- It is not known how the name "*knapsack problem*" originated, though the problem was referred to as such in the early works of mathematician Tobias Dantzig (18841956), suggesting that the name could have existed in folklore before a mathematical problem had been fully defined.

Knapsack Problems

- The most common problem being solved is the **0-1 knapsack problem**, which restricts the number x_i of copies of each kind of item to zero or one. (i.e. Binary).

- Given a set of n items numbered from 1 up to n, each with a weight wi and a value vi, along with a maximum weight capacity W,

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0, 1\}.$$

(Remark: None or One of each item)

- Here x_i represents the number of instances of item i to include in the knapsack.
- Informally, the problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity.
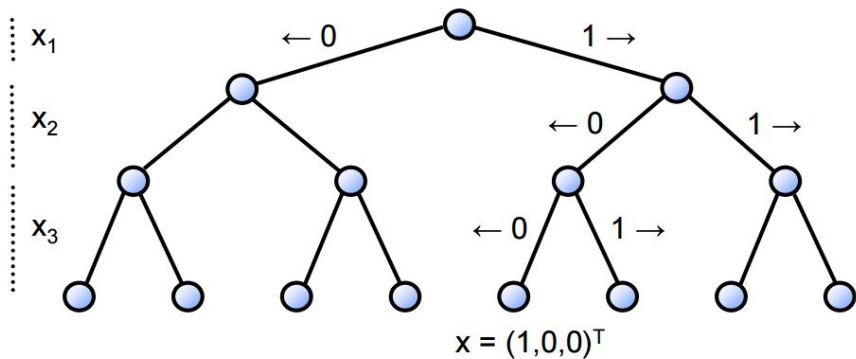
Knapsack Problems

Suppose we have three items as shown in the table below, and suppose the capacity of the knapsack is 5. Formulate the problem.

Item (j)	Weight (w_j)	Benefit(b_j)
1	2	65
2	3	80
3	1	30

Knapsack Problems

- **Remark:** The 0-1 knapsack problems is an optimization problem.
- **Brute force:** Try all 2^n possible subsets (Recall definition of Power Set)
- How many combinations if there are 10 items?
- **Question:** Any solution better than the brute-force?

Knapsack Problems

$x = (1,0,0)^\mathsf{T}$

Branch and Bound

- Relax the $0/1$ integer constraint
- i.e. Let "$x_i \in \{0, 1\}$" become "$0 \le x_i \le 1$".
- Branch on the most fractional variable.

Knapsack Problems

The **bounded knapsack problem (BKP)** removes the restriction that there is only one of each item, but restricts the number x_i of copies of each kind of item to an integer value c_i:

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0, \ldots, c_i\}$$

Knapsack Problems

The **unbounded knapsack problem (UKP)** places no upper bound on the number of copies of each kind of item and can be formulated as above except for that the only restriction on x_i is that it is a non-negative integer.

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \geq 0$$