0-1 Knapsack problem

0-1 Knapsack problem

Given a knapsack with maximum capacity W, and a set S consisting of n items

Each item i has some weight w_i and benefit value b_i (all w_i, b_i and W are integer values)

Problem: How to pack the knapsack to achieve maximum total value of packed items?

0-1 Knapsack problem

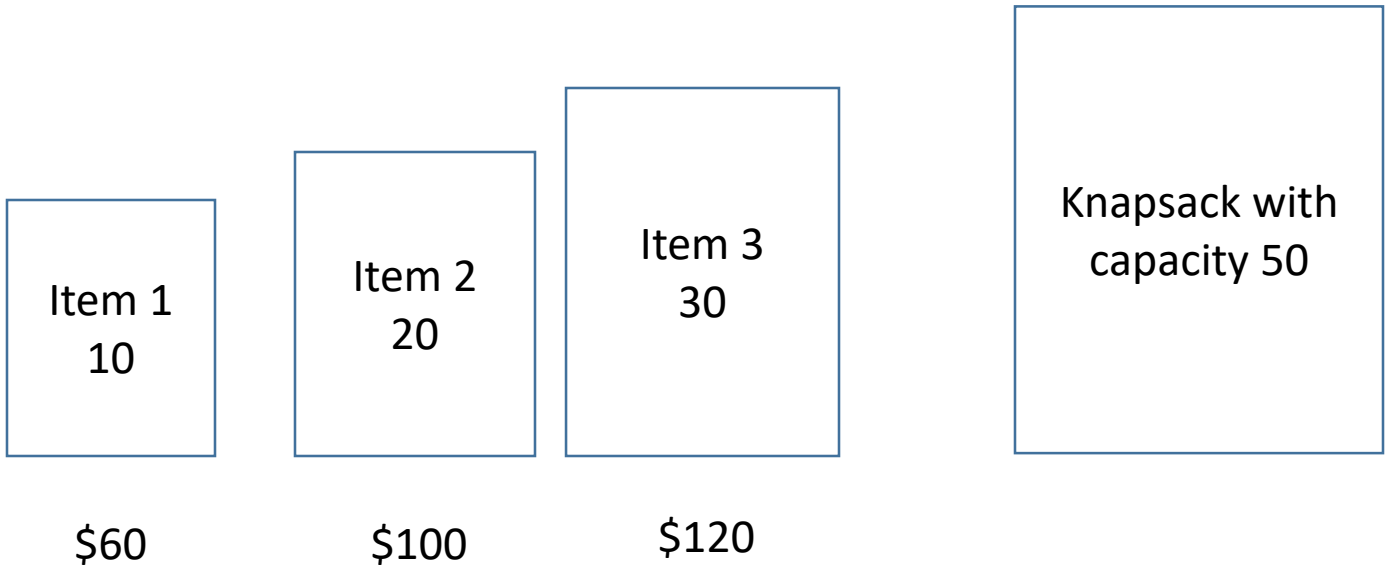
Problem, in other words, is to find

$$\max \sum_{i \in T} b_i \text{ subject to } \sum_{i \in T} w_i \leq W$$

The problem is called a *"0-1"* problem, because each item must be entirely accepted or rejected.

Just another version of this problem is the "*Fractional Knapsack Problem*", where we can take fractions of items.

0-1 Knapsack problem (Example)



0-1 Knapsack problem: brute- force approach

Let's first solve this problem with a straightforward algorithm

Since there are *n* items, how many possible combination of items are possible?

It is the same problem as determining all possible subsets of a set.

0-1 Knapsack problem: brute- force approach

Since there are n items, there are 2^n possible combinations of items.

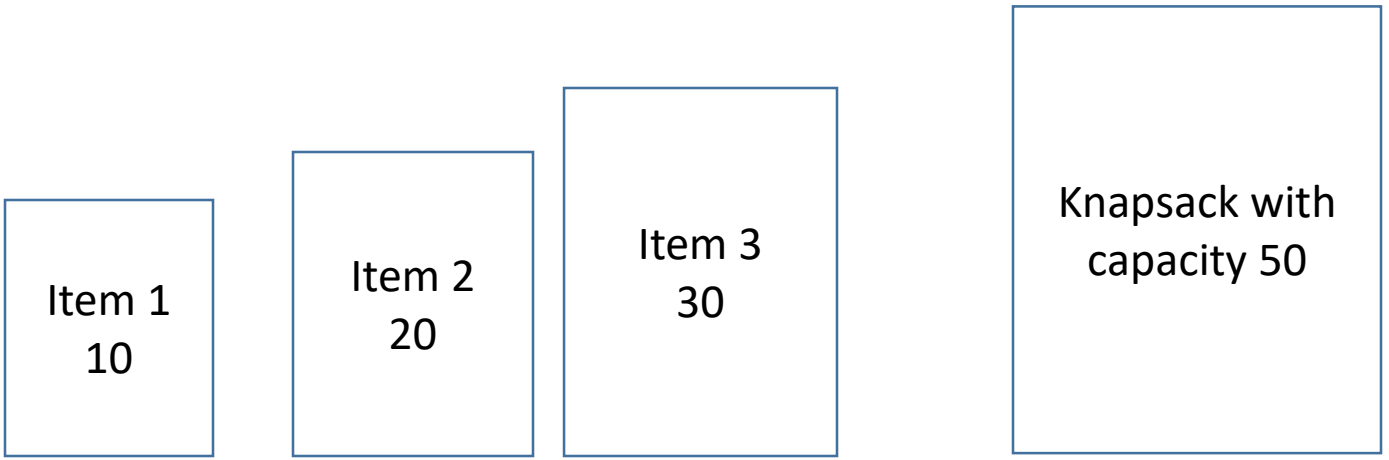
We go through all combinations and find the one with the most total value and with total weight less or equal to W

Running time will be $O(2^n)$

Does any greedy strategy work for 0-1 knapsack problem?

Lets find out!

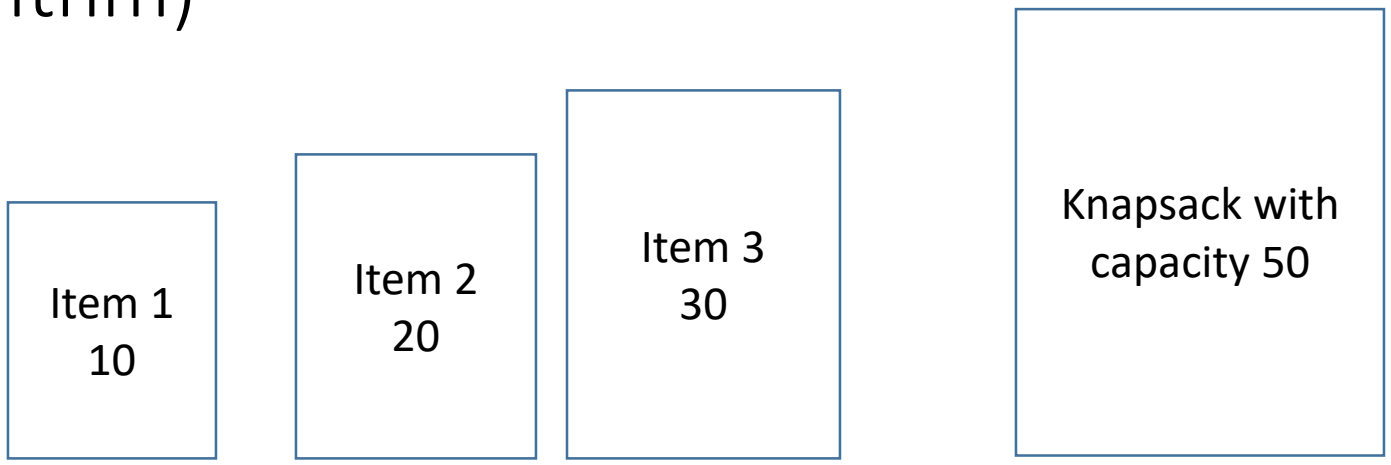
0-1 Knapsack problem (Example 1)



Value per weight: Item 1 (6) Item 2 (5) Item 3 (4)

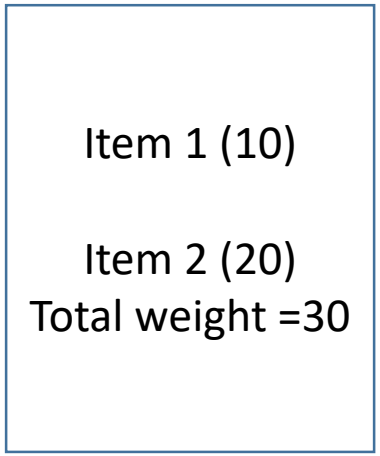
Lets try value per weight greedy strategy that we used for fractional knapsack

0-1 Knapsack (Counter Example for Greedy Algorithm)



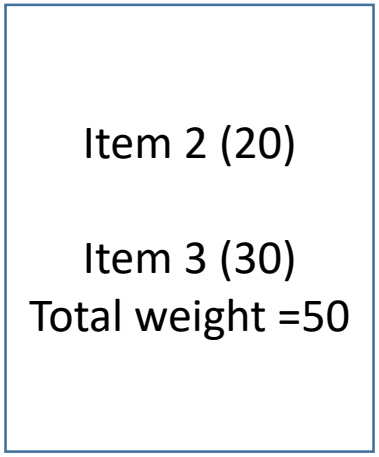
Value per weight: Item 1 (6) Item 2 (5) Item 3 (4)

Value per weight Greedy Solution



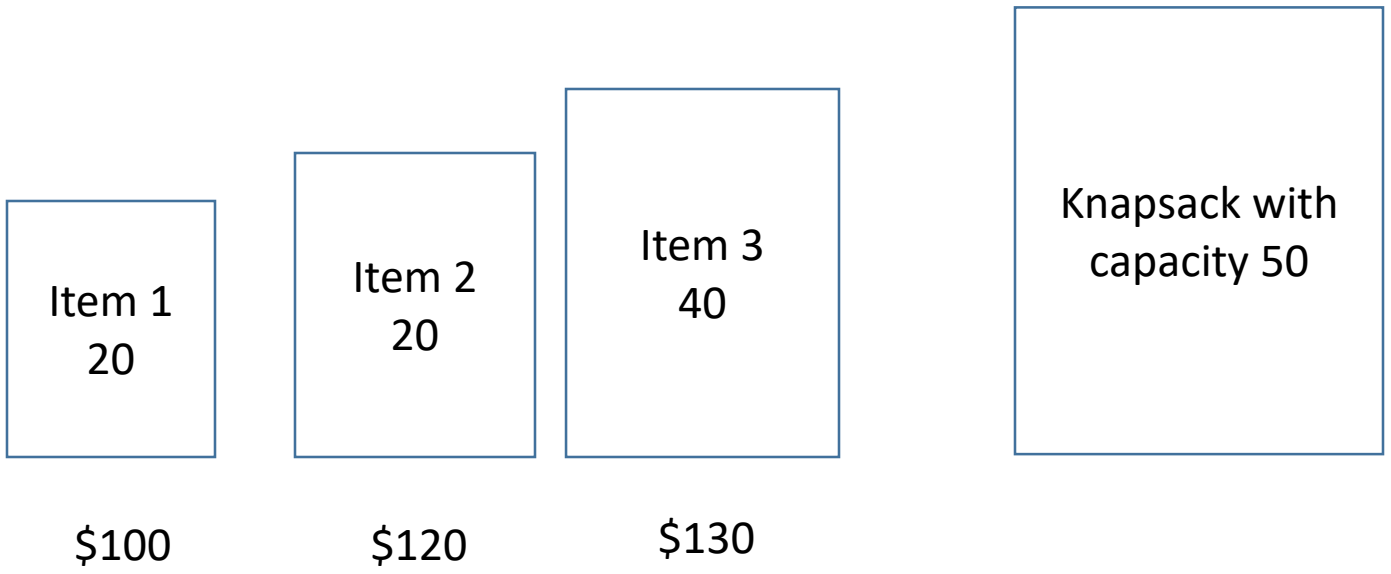
Total Value = $160

Optimal Solution



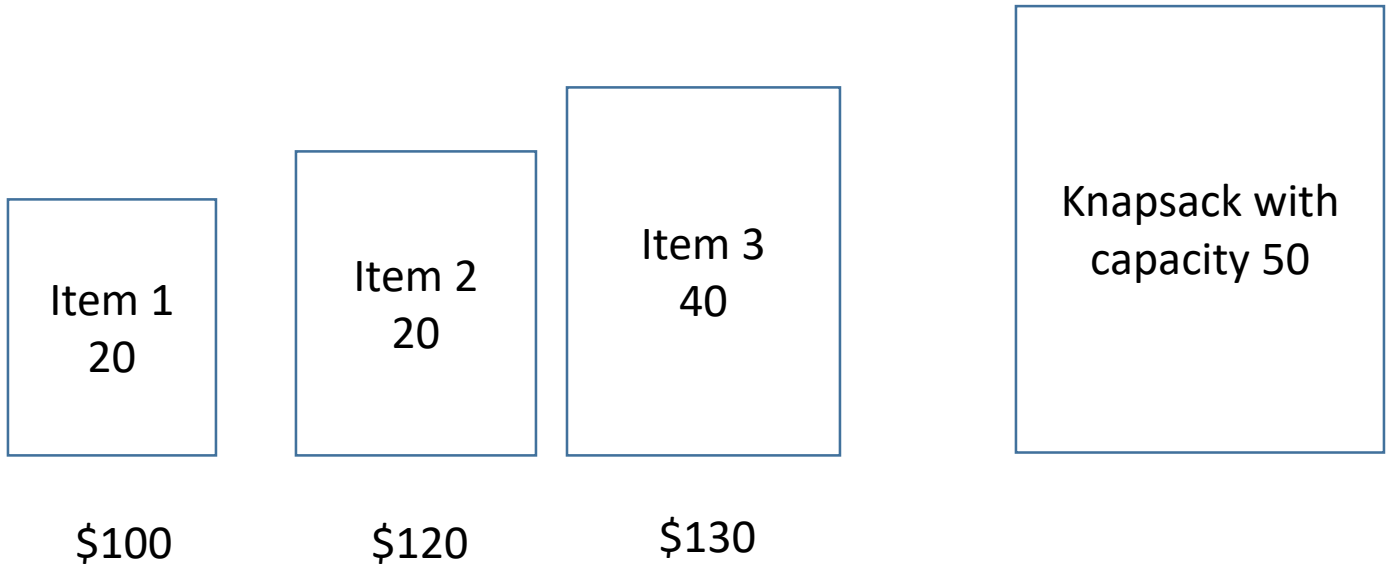
Total Value = $220

0-1 Knapsack problem (Example 2)

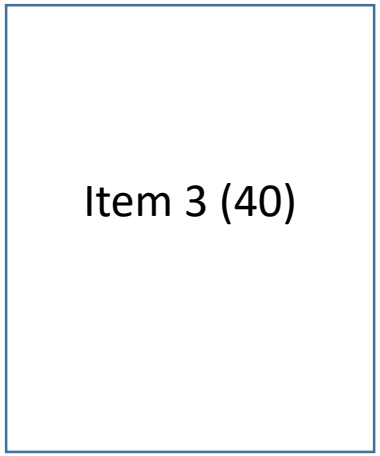


Lets try greedy algorithm: Select item with maximum value first.

0-1 Knapsack problem (Example 2)

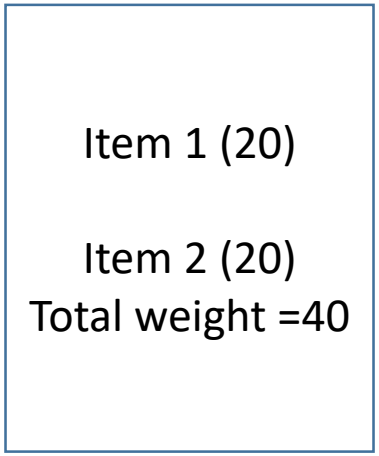


Maximum value first
Greedy Solution



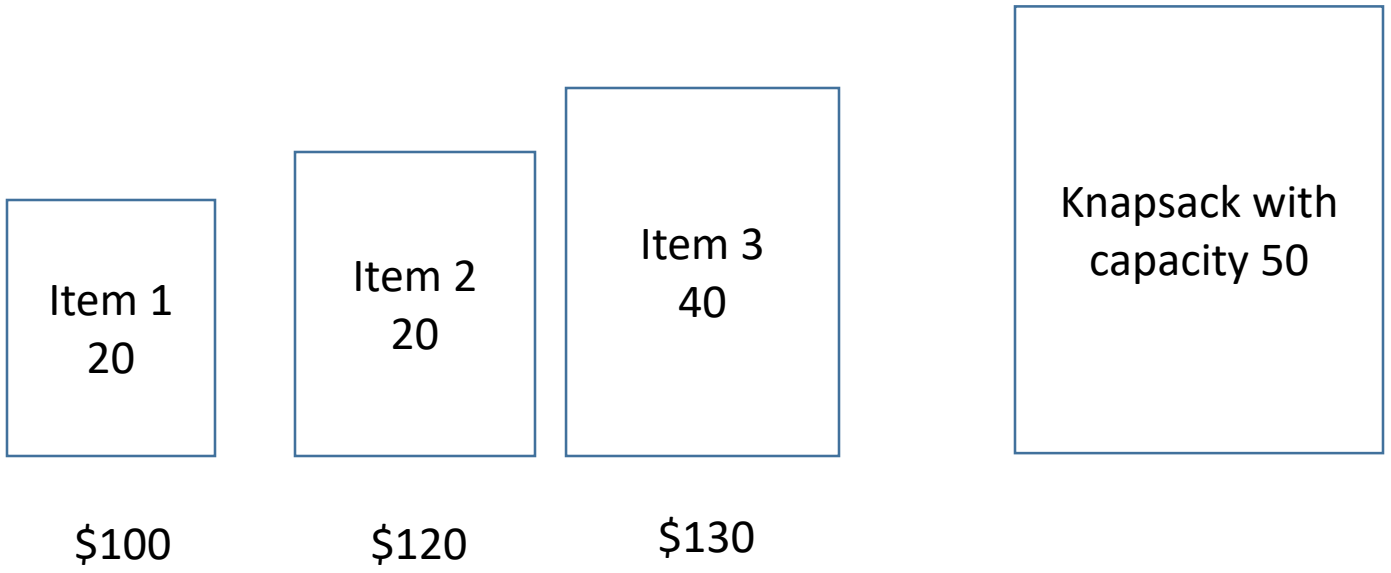
Total Value = $130

Optimal
Solution



Total Value = $220

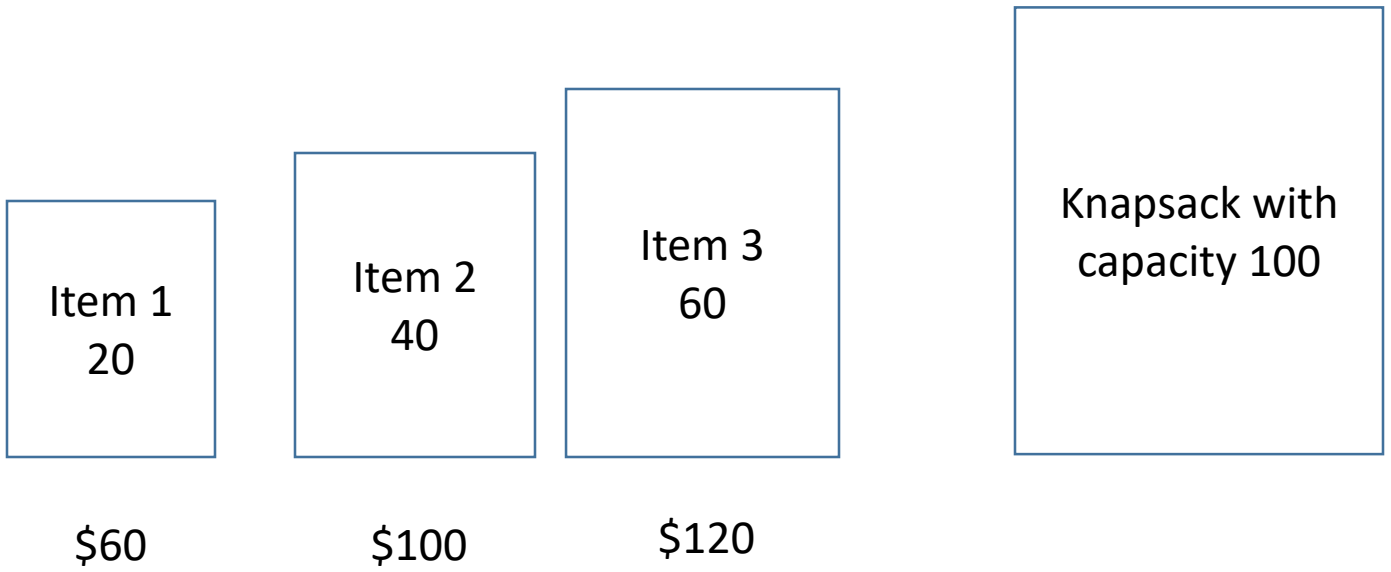
0-1 Knapsack problem (Example 2)



Lets try another greedy algorithm: Select item with minimum weight first.

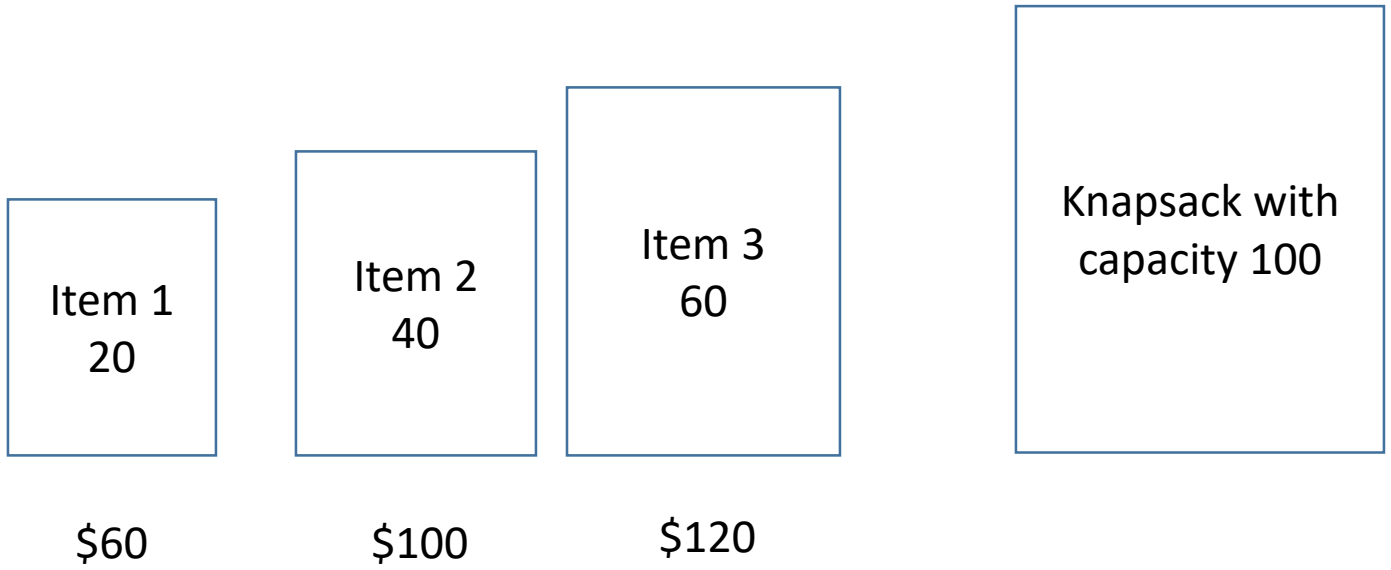
It will give optimal solution in this example with value $220

0-1 Knapsack problem (Example 3)

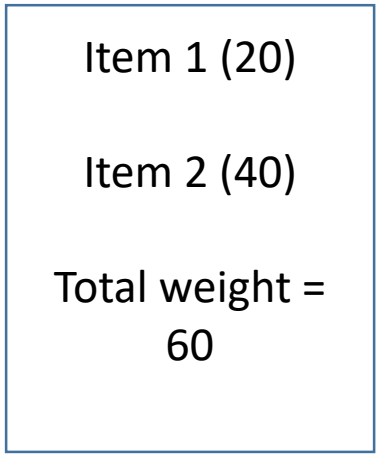


Lets try greedy algorithm: Select item with minimum weight first.

0-1 Knapsack problem (Example 3)

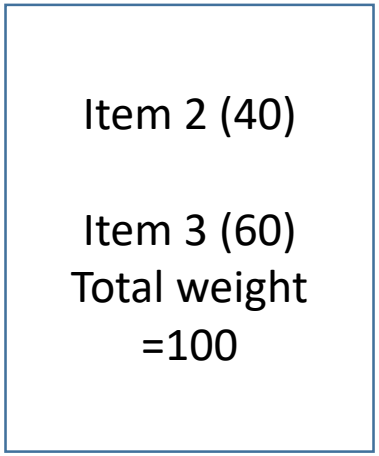


Minimum weight first Greedy Solution



Total Value = $160

Optimal Solution



Total Value = $220

No greedy strategy guarantees optimal solution for 0-1 Knapsack problem

- We have seen counter examples for three greedy algorithms in previous slides