

Assignment #3

**Deadline: Tuesday, November 26, 2020
before 10:00 AM (to be delivered in BrightSpace)**

In class, we studied the $[0, 1]$ -knapsack problem. In this assignment you have to explain how to solve the $\{0, 1\}$ -knapsack problem using dynamic programming.

You are given n objects which *cannot be broken into smaller pieces*. Moreover, *you have only one copy of each object*. Each object i (where $1 \leq i \leq n$) has an integer weight $w_i > 0$ and a value $v_i > 0$. You have a knapsack that can carry a total weight not exceeding W . Your goal is to fill the knapsack in a way that maximizes the total value of the included objects, while respecting the capacity constraint. For each object i (where $1 \leq i \leq n$), either you bring it or not.

1. (10 marks) Write a recursion for the optimal solution and explain why it is correct. **Make sure you define the notation you are using.**

Solution: For $0 \leq i \leq n$ and $0 \leq j \leq W$, let $V[i, j]$ be the maximum value of the objects we can carry if the weight limit is j and we are only allowed to include objects numbered from 1 to i . For all $0 \leq i \leq n$, we have $V[i, 0] = 0$ since the weight limit is 0. For all $0 \leq j \leq W$, we have $V[0, j] = 0$ since no objects are allowed.

Suppose we know all values up to $V[i - 1, j]$ and consider the object i . Either the object i does not help improving the total value of the objects we are carrying or it does help. If it does not help, then the solution was already optimal with the first $i - 1$ objects. In other words, we have $V[i, j] = V[i - 1, j]$. If it does help, then we must include it, so we get $V[i, j] = V[i - 1, j - w_i] + v_i$. We only need to be careful: for this expression to make sense, we need to have $j - w_i \geq 0$. Therefore, the recursion for the optimal solution is

$$\begin{aligned} V[i, 0] &= 0 && 0 \leq i \leq n \\ V[0, j] &= 0 && 0 \leq j \leq W \\ V[i, j] &= V[i - 1, j] && \text{if } w_i > j, \\ V[i, j] &= \max\{V[i - 1, j], V[i - 1, j - w_i] + v_i\} && \text{otherwise.} \end{aligned}$$

2. (10 marks) Consider the following input and fill the table corresponding to the recursion you found in #1: $n = 6$, $w_1 = 2$, $w_2 = 6$, $w_3 = 5$, $w_4 = 8$, $w_5 = 4$, $w_6 = 9$, $v_1 = 1$, $v_2 = 2$, $v_3 = 1$, $v_4 = 3$, $v_5 = 2$, $v_6 = 4$ and $W = 15$. Moreover, give all optimal solutions.

Solution:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------------------|----------|---|----------|---|---|---|----------|---|---|---|----|----|----|----|----|----------|
| $i = 0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $w_1 = 2$ and $v_1 = 1$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $w_2 = 6$ and $v_2 = 2$ | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| $w_3 = 5$ and $v_3 = 1$ | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
| $w_4 = 8$ and $v_4 = 3$ | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 |
| $w_5 = 4$ and $v_5 = 2$ | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 6 | 6 |
| $w_6 = 9$ and $v_6 = 4$ | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 6 | 6 | 7 |

The only optimal solution is $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 0$, $x_5 = 1$ and $x_6 = 1$. In other words, we carry objects 1, 5 and 6 in the knapsack for a total weight of 15 and a total value of 7.