



# Knapsack Problem

Nombre del Alumno: Jorge Alejandro González Guerra

Matrícula: 1889169

Carrera: Ingeniería en Tecnologías de la Información (*ITS*)

Maestro: Iris Abril Martínez Salazar

Curso: Temas Selectos de Optimización

Hora: Jueves M4-M6

Fecha de Entrega: 28 de Mayo de 2020

## Description:

It is required to test 3 different strategies to solve a Knapsack problem:

1. Considering Most Valuable Items first.
2. Considering Items with Less Weight first.
3. Considering Value/Weight relation, with the better relations first.

Our knapsack has a Maximum capacity of 100 (weight carry capacity), and we're provided with 35 items to test:

Item	Value	Weight
1	36	15
2	32	19
3	31	13
4	12	8
5	43	15
6	12	16
7	18	6
8	47	22
9	29	12
10	31	25
11	41	19
12	33	11
13	26	21
14	20	10
15	23	23
16	41	20
17	44	16
18	46	5
19	47	16
20	20	7
21	48	14
22	17	21
23	45	25
24	33	9
25	14	17

26	16	21
27	41	20
28	36	9
29	29	21
30	30	8
31	34	13
32	12	22
33	14	8
34	30	7

## First Strategy: Most Valuable First

Our results were not what we expected. We ended up adding way less items than initially predicted:

```

**** Add using Value strategy (Most Valuable First)...
Adding Item 23 (Value: 25)    →    Capacity: 100 - 45 = 55 ✓
Adding Item 10 (Value: 25)    →    Capacity: 55 - 31 = 24 ✓
Adding Item 15 (Value: 23)    →    Capacity: 24 - 23 = 1  ✓
Adding Item 32 (Value: 22)    →    Capacity: 1 - 12 = -11 ✗
Adding Item 8 (Value: 22)     →    Capacity: 1 - 47 = -46 ✗
Adding Item 29 (Value: 21)    →    Capacity: 1 - 29 = -28 ✗
Adding Item 26 (Value: 21)    →    Capacity: 1 - 16 = -15 ✗
Adding Item 13 (Value: 21)    →    Capacity: 1 - 26 = -25 ✗
Adding Item 22 (Value: 21)    →    Capacity: 1 - 17 = -16 ✗
Adding Item 16 (Value: 20)    →    Capacity: 1 - 41 = -40 ✗
Adding Item 27 (Value: 20)    →    Capacity: 1 - 41 = -40 ✗
Adding Item 11 (Value: 19)    →    Capacity: 1 - 41 = -40 ✗
Adding Item 2 (Value: 19)     →    Capacity: 1 - 32 = -31 ✗
Adding Item 25 (Value: 17)    →    Capacity: 1 - 14 = -13 ✗
Adding Item 6 (Value: 16)     →    Capacity: 1 - 12 = -11 ✗
Adding Item 19 (Value: 16)    →    Capacity: 1 - 47 = -46 ✗
Adding Item 17 (Value: 16)    →    Capacity: 1 - 44 = -43 ✗
Adding Item 1 (Value: 15)     →    Capacity: 1 - 36 = -35 ✗
Adding Item 5 (Value: 15)     →    Capacity: 1 - 43 = -42 ✗
Adding Item 21 (Value: 14)    →    Capacity: 1 - 48 = -47 ✗
Adding Item 31 (Value: 13)    →    Capacity: 1 - 34 = -33 ✗
Adding Item 3 (Value: 13)     →    Capacity: 1 - 31 = -30 ✗
Adding Item 9 (Value: 12)     →    Capacity: 1 - 29 = -28 ✗
Adding Item 12 (Value: 11)    →    Capacity: 1 - 33 = -32 ✗
Adding Item 14 (Value: 10)    →    Capacity: 1 - 20 = -19 ✗
Adding Item 24 (Value: 9)     →    Capacity: 1 - 33 = -32 ✗
Adding Item 28 (Value: 9)     →    Capacity: 1 - 36 = -35 ✗
Adding Item 33 (Value: 8)     →    Capacity: 1 - 14 = -13 ✗
Adding Item 4 (Value: 8)      →    Capacity: 1 - 12 = -11 ✗
Adding Item 30 (Value: 8)     →    Capacity: 1 - 30 = -29 ✗
Adding Item 34 (Value: 7)     →    Capacity: 1 - 30 = -29 ✗
Adding Item 20 (Value: 7)     →    Capacity: 1 - 20 = -19 ✗
Adding Item 7 (Value: 6)      →    Capacity: 1 - 18 = -17 ✗
Adding Item 18 (Value: 5)     →    Capacity: 1 - 46 = -45 ✗
**** Total Items Added: 3

```

## Second Strategy: Less Weight First

Here, our results were aswell not what we expected. We were able to add more items than predicted:

```
**** Add using Weight strategy (Less Weight First)...
```

Adding Item 4	(Weight: 12)	→	Capacity: 100 - 12 = 88	✓
Adding Item 32	(Weight: 12)	→	Capacity: 88 - 12 = 76	✓
Adding Item 6	(Weight: 12)	→	Capacity: 76 - 12 = 64	✓
Adding Item 33	(Weight: 14)	→	Capacity: 64 - 14 = 50	✓
Adding Item 25	(Weight: 14)	→	Capacity: 50 - 14 = 36	✓
Adding Item 26	(Weight: 16)	→	Capacity: 36 - 16 = 20	✓
Adding Item 22	(Weight: 17)	→	Capacity: 20 - 17 = 3	✓
Adding Item 7	(Weight: 18)	→	Capacity: 3 - 18 = -15	✗
Adding Item 20	(Weight: 20)	→	Capacity: 3 - 20 = -17	✗
Adding Item 14	(Weight: 20)	→	Capacity: 3 - 20 = -17	✗
Adding Item 15	(Weight: 23)	→	Capacity: 3 - 23 = -20	✗
Adding Item 13	(Weight: 26)	→	Capacity: 3 - 26 = -23	✗
Adding Item 29	(Weight: 29)	→	Capacity: 3 - 29 = -26	✗
Adding Item 9	(Weight: 29)	→	Capacity: 3 - 29 = -26	✗
Adding Item 30	(Weight: 30)	→	Capacity: 3 - 30 = -27	✗
Adding Item 34	(Weight: 30)	→	Capacity: 3 - 30 = -27	✗
Adding Item 10	(Weight: 31)	→	Capacity: 3 - 31 = -28	✗
Adding Item 3	(Weight: 31)	→	Capacity: 3 - 31 = -28	✗
Adding Item 2	(Weight: 32)	→	Capacity: 3 - 32 = -29	✗
Adding Item 24	(Weight: 33)	→	Capacity: 3 - 33 = -30	✗
Adding Item 12	(Weight: 33)	→	Capacity: 3 - 33 = -30	✗
Adding Item 31	(Weight: 34)	→	Capacity: 3 - 34 = -31	✗
Adding Item 28	(Weight: 36)	→	Capacity: 3 - 36 = -33	✗
Adding Item 1	(Weight: 36)	→	Capacity: 3 - 36 = -33	✗
Adding Item 27	(Weight: 41)	→	Capacity: 3 - 41 = -38	✗
Adding Item 11	(Weight: 41)	→	Capacity: 3 - 41 = -38	✗
Adding Item 16	(Weight: 41)	→	Capacity: 3 - 41 = -38	✗
Adding Item 5	(Weight: 43)	→	Capacity: 3 - 43 = -40	✗
Adding Item 17	(Weight: 44)	→	Capacity: 3 - 44 = -41	✗
Adding Item 23	(Weight: 45)	→	Capacity: 3 - 45 = -42	✗
Adding Item 18	(Weight: 46)	→	Capacity: 3 - 46 = -43	✗
Adding Item 8	(Weight: 47)	→	Capacity: 3 - 47 = -44	✗
Adding Item 19	(Weight: 47)	→	Capacity: 3 - 47 = -44	✗
Adding Item 21	(Weight: 48)	→	Capacity: 3 - 48 = -45	✗

```
**** Total Items Added: 7
```

## Third Strategy: Value/Weight Relation

Our results were almost those that we expected. Enough values were added to the knapsack (considering the overall weight) but we ended up adding a little less items than with the second strategy:

```
**** Add using Value/Weight strategy...
Adding Item 32 (Value/Weight: 1.8333) → Capacity: 100 - 12 = 88 ✓
Adding Item 6 (Value/Weight: 1.3333) → Capacity: 88 - 12 = 76 ✓
Adding Item 26 (Value/Weight: 1.3125) → Capacity: 76 - 16 = 60 ✓
Adding Item 22 (Value/Weight: 1.2353) → Capacity: 60 - 17 = 43 ✓
Adding Item 25 (Value/Weight: 1.2143) → Capacity: 43 - 14 = 29 ✓
Adding Item 15 (Value/Weight: 1.0000) → Capacity: 29 - 23 = 6 ✓
Adding Item 13 (Value/Weight: 0.8077) → Capacity: 6 - 26 = -20 ✗
Adding Item 10 (Value/Weight: 0.8065) → Capacity: 6 - 31 = -25 ✗
Adding Item 29 (Value/Weight: 0.7241) → Capacity: 6 - 29 = -23 ✗
Adding Item 4 (Value/Weight: 0.6667) → Capacity: 6 - 12 = -6 ✗
Adding Item 2 (Value/Weight: 0.5938) → Capacity: 6 - 32 = -26 ✗
Adding Item 33 (Value/Weight: 0.5714) → Capacity: 6 - 14 = -8 ✗
Adding Item 23 (Value/Weight: 0.5556) → Capacity: 6 - 45 = -39 ✗
Adding Item 14 (Value/Weight: 0.5000) → Capacity: 6 - 20 = -14 ✗
Adding Item 16 (Value/Weight: 0.4878) → Capacity: 6 - 41 = -35 ✗
Adding Item 27 (Value/Weight: 0.4878) → Capacity: 6 - 41 = -35 ✗
Adding Item 8 (Value/Weight: 0.4681) → Capacity: 6 - 47 = -41 ✗
Adding Item 11 (Value/Weight: 0.4634) → Capacity: 6 - 41 = -35 ✗
Adding Item 3 (Value/Weight: 0.4194) → Capacity: 6 - 31 = -25 ✗
Adding Item 1 (Value/Weight: 0.4167) → Capacity: 6 - 36 = -30 ✗
Adding Item 9 (Value/Weight: 0.4138) → Capacity: 6 - 29 = -23 ✗
Adding Item 31 (Value/Weight: 0.3824) → Capacity: 6 - 34 = -28 ✗
Adding Item 17 (Value/Weight: 0.3636) → Capacity: 6 - 44 = -38 ✗
Adding Item 20 (Value/Weight: 0.3500) → Capacity: 6 - 20 = -14 ✗
Adding Item 5 (Value/Weight: 0.3488) → Capacity: 6 - 43 = -37 ✗
Adding Item 19 (Value/Weight: 0.3404) → Capacity: 6 - 47 = -41 ✗
Adding Item 12 (Value/Weight: 0.3333) → Capacity: 6 - 33 = -27 ✗
Adding Item 7 (Value/Weight: 0.3333) → Capacity: 6 - 18 = -12 ✗
Adding Item 21 (Value/Weight: 0.2917) → Capacity: 6 - 48 = -42 ✗
Adding Item 24 (Value/Weight: 0.2727) → Capacity: 6 - 33 = -27 ✗
Adding Item 30 (Value/Weight: 0.2667) → Capacity: 6 - 30 = -24 ✗
Adding Item 28 (Value/Weight: 0.2500) → Capacity: 6 - 36 = -30 ✗
Adding Item 34 (Value/Weight: 0.2333) → Capacity: 6 - 30 = -24 ✗
Adding Item 18 (Value/Weight: 0.1087) → Capacity: 6 - 46 = -40 ✗
**** Total Items Added: 6
```

## **So, which strategy turned out to work better?**

The *Less Weight First* strategy ended up working better for this scenario. I believe that the *Value/Weight Relation* strategy can work better in some other scenarios where the knapsack has a higher capacity, or the overall value and weight of the items are not that similar, a better categorization can take place with the third strategy.