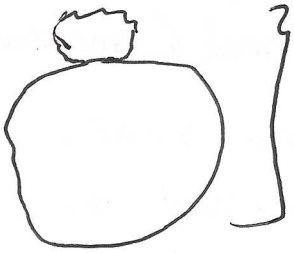
- Now let us focus our attention back to the 0/1 Knapsack problem. As a first step we will perform an exercise to identify that this problem can be solved using Dynamic Programming.

"How does one identify that DP can be applied to the 0/1 Knapsack problem?"

Let us first take a look at what input has been provided to us to [redacted] in order for us to solve the problem.

Input :



A bag whose capacity is provided as a variable "W"

{ O O O O O ... } a collection of items whose capacities and value in USD is provided as individual arrays: w[], p[]

hence our input is the following for 0/1 KS problem :

{ W
w[]
p[] }

the sample instance of the input we will take a look at here is the following:

W : 7 Kg

w[] : 1 3 4 5

p[] : 1 4 5 7

Output : The output we want to have given an input instance is the "maximum profit/value" one could obtain by putting a subset of items in the bag without exceeding the knapsack capacity.