1 Introduction

Reprenons le problème du sac à dos. On possède n objets O_i, $1 \leq i \leq n$ de valeur v_i et de masse p_i. On souhaite choisir les objets de façon à maximiser la valeur du sac sans dépasser la masse maximum du sac P.

— Ce problème possède bien des sous-structures optimales : si, avec un choix de n objets, on place un objet O_0 de masse p_0 dans le sac acceptant un chargement de masse maximum P, le problème revient à déterminer le chargement de plus grande valeur avec les $n-1$ objets restants pour une masse maximum de $P - p_0$.

— Par ailleurs, il y a bien chevauchement de sous-problèmes car si le choix d'un premier objet O_i nécessite d'étudier l'ajout d'un second objet O_j, de même un choix de O_j en premier doit nécessiter l'étude de l'ajout de O_i.

Notons $V(i, P)$ la valeur optimale du problème du sac à dos pour les objets et une masse maximale P. Cette grandeur vérifie la relation de récurrence suivante :

$$V(i,P) = \max \begin{cases} V(i-1,P) \\ v_i + V(i-1, P-p_i) \ \# \text{ On ajoute un l'objet i} \end{cases}$$

avec les conditions initiales : $V(0,p) = 0$ et $V(i,0) = 0$.

2 Tableau des valeurs

Prenons $P = 7$ et l'ensemble des objets :

$$\{(3,2),(1,2),(3,4),(4,5),(2,3)\} \text{ (masse, valeurs)}$$

Nous construisons alors le tableau suivant :

k	p_i	v_i	$O_i \setminus P$	0	1	2	3	4	5	6	7
0	0	0	pas d'objet	0	0	0	0	0	0	0	0
1	3	2	O_1	0	0	0	2	2	2	2	2
2	1	2	O_2	0	2	2	2	4	4	4	4
3	3	4	O_3	0	2	2	4	6	6	6	8
4	4	5	O_4	0	2	2	4	6	7	7	9
5	2	3	O_5								

TABLE 1 – Tableau des valeurs $V(k,p)$

1. Avec la relation de récurrence, remplir la dernière ligne du tableau et en déduire la valeur maximale du sac pour $k = 5$ et $P = 7$.

2. Maintenant que la valeur optimale a été obtenue, on veut en déduire la solution optimale, c ?est à dire la composition du sac à dos de plus grande valeur. Il suffit pour cela de "remonter les résultats du tableau". Par exemple, la valeur de $V(5,7)$ est soit égale à $V(4,7)$, soit égale à $V(4, 7-p5) + v5 = V(4,5) + 3$. Comme seule la seconde égalité est vraie, alors $O5$ appartient à la solution optimale. Déterminer les objets contenus dans la solution optimale.

3 Implémentation en Python

Le code suivant permet de résoudre le problème :

```
# dictionnaire des objets {numero: [masse, valeurs]}
Objets = {1:[3,2],2:[1,2], 3:[3,4], 4:[4,5], 5:[2,3] }

# initialisation du tableau des valeurs
n = len(Objets)
Pmax = 7
T=[[0]*(Pmax+1) for i in range(n+1)]

#remplissage du tableau
for i in range(1, n+1): # i désigne le numero de l'objet
    for j in range(1, Pmax+1): # j désigne la masse
        if Objets[i][0] > j:  # la contrainte de masse maximale
            T[i][j] = ............
        else:
            a = ............
            b= Objets[i][1] + T[i-1][j-Objets[i][0]]
            T[i][j]=max(a,b)

#affichage du tableau
print(T)

# reconstruction d'une solution optimale
sol = []
k , p = n, Pmax  # initialisation : on part du coin inférieur droit
while k > 0:
    if T[k][p] != T[k-1][p]:
        sol.append(.........)
        p = ..........
    k -= 1

print(sol)
```