Informatique - Chapitre 3
Programmation dynamique

Le problème du sac à dos : code et résultats

```python
import numpy as np

# Problème du sac à dos : algorithme glouton par valeur simple
def sacados_glouton_simple(valeurs: [int], masses: [int], Masse_max: int) -> int:
    # Récupération des objets
    objets = []
    for i in range(len(valeurs)):
        objets.append( [valeurs[i], masses[i]] )
    # Tri des objets par valeur décroissante (tri selon le premier élément)
    objets = sorted(objets, reverse = True)
    print(" Liste des objets classés :", objets)
    # Remplissage du sac
    valeur = 0
    objets_sac = []
    for objet in objets:
        if objet[1] <= Masse_max:
            valeur = valeur + objet[0]
            Masse_max = Masse_max - objet[1]
            objets_sac.append(objet)
    return valeur, objets_sac

# Problème du sac à dos : algorithme glouton par valeur massique
def sacados_glouton_massic(valeurs: [int], masses: [int], Masse_max: int) -> int:
    # Récupération des objets
    objets = []
    for i in range(len(valeurs)):
        objets.append( [valeurs[i], masses[i]] )
    # Tri des objets par valeur massique décroissante
    def valeurmassique(objet):
        return objet[0] / objet[1]
    objets = sorted(objets, key = valeurmassique, reverse = True)
    print(" Liste des objets classés :", objets)
    # Remplissage du sac
    valeur = 0
    objets_sac = []
    for objet in objets:
        if objet[1] <= Masse_max:
            valeur = valeur + objet[0]
            Masse_max = Masse_max - objet[1]
            objets_sac.append(objet)
    return valeur, objets_sac
```

```python
# Problème du sac à dos : algorithme dynamique avec tableau
def sacados_dynamique_tab(valeurs: [int], masses: [int], Masse_max: int) -> int:
    n = len(valeurs)
    # Remplissage du tableau de valeurs optimales
    s = np.zeros( (n+1, Masse_max+1), dtype = int)
    for k in range(1, n+1):
        for M in range(1, Masse_max+1):
            if masses[k-1] <= M:
                s[k, M] = max( s[k-1,M], s[k-1,M-masses[k-1]] + valeurs[k-1] )
            else:
                s[k, M] = s[k-1, M]
    # Affichage du tableau des valeurs calculées
    print(" Tableau des valeurs calculées, de taille",(n+1), "*", (Masse_max+1))
    print(s)
    # Récupération des différents objets
    M = Masse_max
    objets = []
    for k in range(n, 0, -1):
        if s[k-1, M] < s[k, M]:
            objets.append( [ valeurs[k-1], masses[k-1] ] )
            M = M - masses[k-1]
    return s[n, Masse_max], objets

# Problème du sac à dos : algorithme dynamique avec dictionnaire
def sacados_dynamique_dico(valeurs: [int], masses: [int], Masse_max: int) -> int:
    s = {}
    n = len(valeurs)
    # Fonction récursive appliquant l'équation de Bellman
    def calcul(k, M):
        if (k,M) not in s:
            if k == 0 or M == 0:
                x = 0
            elif masses[k-1] <= M:
                x = max(calcul(k-1,M), calcul(k-1,M-masses[k-1]) + valeurs[k-1])
            else:
                x = calcul(k-1,M)
            s[ (k,M) ] = x
            #print(" Nouveau calcul avec k =",k,"et M =",M,": s =", s[(k,M)])
        #else:
        #   print(" Pas de modification pour k =",k,"et M =",M,": s =",s[(k,M)])
        return s[ (k,M) ]
    # Calcul
    val = calcul(n, Masse_max)
    # Affichage du dictionnaire des valeurs calculées
    print(" Dictionnaire des valeurs calculées, de taille",len(s))
    print(s)
    # Récupération des différents objets
    M = Masse_max
    objets = []
    for k in range(n, 0, -1):
        if s[ (k-1,M) ] < s[ (k,M) ]:
            objets.append([ valeurs[k-1], masses[k-1] ])
            M = M - masses[k-1]
    return val, objets
```

```
# Problème du sac à dos : résolution d'un exemple et comparaison des méthodes
valeurs = [ 12, 4, 10, 6, 8, 3 ]
masses =  [  6, 2,  4, 2, 3, 2 ]
masse_sac = 8
print("*** Algorithme glouton par valeur simple")
s, objets = sacados_glouton_simple(valeurs, masses, masse_sac)
print(" Liste des objets sélectionnés : ", objets)
print(" Valeur totale dans le sac : ", s, "\n")
print("*** Algorithme glouton par valeur massique")
s, objets = sacados_glouton_massic(valeurs, masses, masse_sac)
print(" Liste des objets sélectionnés : ", objets)
print(" Valeur totale dans le sac : ", s, "\n")
print("*** Algorithme dynamique avec tableau")
s, objets = sacados_dynamique_tab(valeurs, masses, masse_sac)
print(" Liste des objets sélectionnés : ", objets)
print(" Valeur totale dans le sac : ", s, "\n")
print("*** Algorithme dynamique avec dictionnaire")
s, objets = sacados_dynamique_dico(valeurs, masses, masse_sac)
print(" Liste des objets sélectionnés : ", objets)
print(" Valeur totale dans le sac : ", s)
```

Résultats

```
*** Algorithme glouton par valeur simple
Liste des objets classés : [[12, 6], [10, 4], [8, 3], [6, 2], [4, 2], [3, 2]]
Liste des objets sélectionnés :  [[12, 6], [6, 2]]
Valeur totale dans le sac :  18

*** Algorithme glouton par valeur massique
Liste des objets classés : [[6, 2], [8, 3], [10, 4], [12, 6], [4, 2], [3, 2]]
Liste des objets sélectionnés :  [[6, 2], [8, 3], [4, 2]]
Valeur totale dans le sac :  18

*** Algorithme dynamique avec tableau
Tableau des valeurs calculées, de taille 7 * 9
[[ 0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0 12 12 12]
[ 0  0  4  4  4  4 12 12 16]
[ 0  0  4  4 10 10 14 14 16]
[ 0  0  6  6 10 10 16 16 20]
[ 0  0  6  8 10 14 16 18 20]
[ 0  0  6  8 10 14 16 18 20]]
Liste des objets sélectionnés :  [[6, 2], [10, 4], [4, 2]]
Valeur totale dans le sac :  20

*** Algorithme dynamique avec dictionnaire
Dictionnaire des valeurs calculées, de taille 37
{(0, 8): 0, (0, 2): 0, (1, 8): 12, (0, 6): 0, (0, 0): 0, (1, 6): 12, (2, 8): 16,
 (0, 4): 0, (1, 4): 0, (1, 2): 0, (2, 4): 4, (3, 8): 16, (2, 6): 12, (1, 0): 0,
 (2, 2): 4, (3, 6): 14, (4, 8): 20, (0, 5): 0, (1, 5): 0, (0, 3): 0, (1, 3): 0,
 (2, 5): 4, (0, 1): 0, (1, 1): 0, (2, 1): 0, (3, 5): 10, (2, 3): 4, (3, 3): 4,
 (4, 5): 10, (5, 8): 20, (2, 0): 0, (3, 4): 10, (4, 6): 16, (3, 1): 0, (4, 3): 6,
 (5, 6): 16, (6, 8): 20}
Liste des objets sélectionnés :  [[6, 2], [10, 4], [4, 2]]
Valeur totale dans le sac :  20
```