



# Cours - NSI

## Algorithmes Gloutons

### Problème du sac à dos

## 1 Énoncé général

On dispose de plusieurs objets possédant chacun un poids et une valeur, et d'un sac à dos acceptant un poids maximum. Quels objets faut-il mettre dans le sac à dos de manière à maximiser la valeur totale sans dépasser le poids maximum autorisé?

On souhaite répondre au problème en utilisant un algorithme glouton.

1. On va donc considérer un critère global : le poids du sac à dos qui ne doit pas dépasser une certaine valeur.
2. On doit choisir aussi un des critères locaux que l'on choisit au départ pour résoudre le problème :
  - Prendre l'objet qui a la plus grande valeur;
  - OU Prendre l'objet qui a le plus petit poids;
  - OU Prendre l'objet qui a le rapport valeur/poids le plus grand.

## 2 Un cas particulier

On possède 4 objets dont les valeurs et masses respectives sont données ci-dessous :

objet n°	1	2	3	4
valeur (en €)	7	4	3	3
masse (en kg)	13	12	10	8

Si l'on dispose d'un sac à dos pouvant accepter une masse maximale de 30 kg, quels objets doit-on choisir? Quelle est la valeur totale? Quelle est la masse totale contenue dans le sac?

1. Écrire une fonction `sac_a_dos` prenant en paramètre la masse maximale du sac à dos et une liste de couple (valeur, poids) . Cette fonction renvoie la liste des objets, la valeur totale des objets et la masse du sac à dos.

On prendra comme critère local, de **prendre l'objet qui a la plus grande valeur**;

```
# La liste des objets
```

```
liste_objets=[(7,13),(4,12),(3,10),(3,8)]
```

Exemple :

```
>>> sac_a_dos(15,liste_objets)
```

```
([(7, 13)], 7, 13)
```

```
>>> sac_a_dos(22,liste_objets)
```

```
([(7, 13), (3, 8)], 10, 21)
```

2. Écrire une deuxième fonction `sac_a_dos2` qui met en oeuvre le deuxième critère : **Prendre l'objet qui a le plus petit poids**.

```
>>> sac_a_dos2(15,liste_objets)
```

```
([(3, 8)], 3, 8)
```

```
>>> sac_a_dos2(22,liste_objets)
```

```
([(3, 8), (3, 10)], 6, 18)
```

3. Quelles sont les réponses obtenues à l'aide des deux fonctions pour une masse maximale de 30 kg? L'une d'elles est-elle optimale?

*On pourra utiliser un arbre pour déterminer toutes les solutions possibles.*

4. Essayer avec les listes

(a) `liste_objet=[(4, 11), (4, 1), (5, 5), (7, 14), (13, 6),  
(14, 13), (14, 4), (10, 9), (6, 13), (9, 4),  
(4, 9), (7, 8), (6, 6), (9, 6), (8, 5), (11, 12),  
(6, 13), (8, 11), (6, 13)]`

La meilleur solution est

`([(4, 1), (13, 6), (14, 4), (10, 9), (9, 4), (9, 6)], 59, 30)`

(b) `liste_objet=[(11, 9), (1, 1), (14, 1), (9, 1), (13, 13),  
(1, 12), (9, 14), (14, 10), (13, 6), (4, 5),  
(12, 1), (12, 2), (13, 12), (14, 15), (9, 3)]`

La meilleur solution est

`([(1, 1), (14, 1), (9, 1), (14, 10), (13, 6),  
(4, 5), (12, 1), (12, 2), (9, 3)], 88, 30)`