

# Les algorithmes gloutons (2)

## . UN AUTRE CAS CLASSIQUE : LE SAC À DOS

### . PRINCIPE

Le remplissage du sac à dos est la seconde situation classique d'utilisation d'algorithme glouton.

Sa transposition peut supporter de nombreux exemples. Le plus connu : Un cambrioleur remplissant son sac, cherche à maximiser la valeur de son larcin.

Dans tous ces cas, la difficulté vient du fait que leur contenant ne supporte qu'un poids limite (on aurait aussi pu choisir un volume limite).

### . LE REMPLISSAGE D'UNE CLÉ USB

En voici une transposition plus... moderne :

Nous disposons d'une **clé USB** qui est déjà bien remplie et sur laquelle **il ne reste que 6 Go de libre**. Nous souhaitons **copier sur cette clé des fichiers vidéos** pour l'emporter en voyage. Chaque fichier a une taille et chaque vidéo a une durée. La durée n'est pas proportionnelle à la taille car les fichiers sont de format différents, certaines vidéos sont de grande qualité, d'autres sont très compressées.

Nous souhaitons **emporter la plus grande durée de vidéos** sur notre clé USB, pour nous occuper le plus longtemps possible. Le tableau qui suit présente les **fichiers de vidéos disponibles avec les durées données en minutes**.

#### Le choix de la structure de données

Dans un premier temps, on peut se poser la question : comment stocker les données de ce tableau pour une exploitation numérique ?

- Le plus simple est de se limiter à une unique unité pour la taille, en Go par exemple.

- Trouver une structure de donnée adaptée : une table.

- Quel type de table ? Une liste de tuples ou une liste de dictionnaires.

#### L'algorithme glouton à la rescousse

Dans un second temps, appliquons la logique gloutonne à notre situation : trouvons la meilleure solution locale, c'est à dire à chaque étape de remplissage.

La meilleure solution locale consiste à sélectionner la vidéo ayant la durée la plus longue, peu importe sa taille.

- Il faut donc trier notre taille en fonction de la durée des vidéos.

- Parcourir la table, vidéo par vidéo. En commençant par celle de plus longue durée.

- Copier cette vidéo si sa taille est inférieure à l'espace encore disponible sur la clé USB.

Fichier	Durée	Taille
Video 1	114	4,57 Go
Video 2	32	630 Mo
Video 3	20	1,65 Go
Video 4	4	85 Mo
Video 5	18	2,15 Go
Video 6	80	2,71 Go
Video 7	5	320 Mo

#### Le choix du type de boucle pour parcourir la table

Dans le cas présent, il paraît très improbable de remplir notre clé USB exactement à son maximum de capacité. Tant que la clé n'est pas complètement pleine, il faut donc vérifier s'il n'y a pas une vidéo de taille plus petite que l'on pourrait encore ajouter.

Dans le "problème du sac à dos", il paraît donc préférable d'utiliser une boucle POUR car il est très probable qu'il faille parcourir l'ensemble de la table.

### . EXERCICE 1

Voici une proposition d'algorithme utilisant une approche gloutonne pour maximiser la durée des vidéos présentes sur la clé USB.

Pour cet algorithme, on fait le choix d'une table composée d'une liste de tuples :

```
table_videos = [('Video 1', 114, 4.57), ('Video 2', 32, 0.630),  
                ('Video 3', 20, 1.65), ('Video 4', 4, 0.085),  
                ('Video 5', 18, 2.15), ('Video 6', 80, 2.71),  
                ('Video 7', 5, 0.320)]
```

```

VARIABLES
videos : liste de tuples
videos_sur_USB : liste de tuples
espace_libre : entier, espace disponible sur la clé USB

DEBUT
  TRIER videos PAR ORDRE DECROISSANT DE DUREE
  POUR video DANS videos
    SI video[2] <= espace_libre
      AJOUTER LA VALEUR video À videos_sur_USB
      espace_libre ← espace_libre - video[2]
    FIN_SI
  FIN_POUR
FIN

```

1. Implémenter cet algorithme en python
2. Calculer la complexité de cet algorithme

## . ALGORITHME D'OPTIMISATION PAR FORCE BRUTE

### . PRINCIPE DE LA FORCE BRUTE

On a vu que les algorithmes gloutons permettaient d'obtenir une solution souvent satisfaisante mais pas nécessairement optimale (la meilleure possible). Comment trouver LA meilleure solution, la solution optimale ?

Une approche efficace, qui manque toutefois de subtilité, consiste à rechercher l'ensemble des solutions possibles puis à sélectionner la solution qui convient le mieux. On parle alors de méthode "par force brute".

Si le principe est simple (il faut tester tous les cas possibles), La mise en œuvre l'est moins. Comment obtenir tous les cas sans les répéter et sans en oublier un ? Cette question pose la difficulté principale. Une méthode de résolution par force brute

Une méthode astucieuse consiste à...

- écrire des nombres binaires ayant le même nombre de bits que d'objets à sélectionner (ou pas).
- chaque bit correspond à un objet.
- on associe le chiffre 1 à un objet s'il est choisi.
- on associe le chiffre 0 si l'objet n'est pas choisi.

Nous obtenons ainsi toutes une série de nombres binaires.

Un exemple minimaliste : cherchons toutes les combinaisons possibles à partir de 3 objets "à prendre ou à laisser" (ex : 3 fournitures scolaires pour remplir la malle de Harry (cf activité plus loin)).

Comme nous avons des choix à faire sur 3 objets, nous allons créer des nombres binaires à 3 bits.

Voici toutes les combinaisons possibles :

Nous obtenons 8 combinaisons possibles (8 façons différentes de remplir la malle avec ces 3 objets, si on reprend l'exemple avec Harry). Il n'y a pas d'autres possibilités, elle sont toutes là !

On voit qu'il suffit de construire tous les nombres binaires ayant le même nombre de bits que d'objets à sélectionner pour obtenir toutes les combinaisons possibles.

Nombre décimal	Nombre binaire	Choix des objets
0	000	Rien !
1	001	Objet3
2	010	Objet2
3	011	Objet2, objet3
4	100	Objet1
5	101	Objet1, objet3
6	110	Objet1, objet2
7	111	Objet1, objet2, objet3

Pour résoudre le "problème du sac à dos", c'est une méthode radicale mais coûteuse en ressources car il faut envisager les  $2^n$  possibilités lorsqu'on a  $n$  objets à choisir.

## . EXERCICE 2 : UNE APPLICATION DE LA FORCE BRUTE

Reprenons notre situation précédente, dans laquelle il faut choisir parmi 7 fichiers vidéos :

- en maximisant la durée d'écoute totale.
- en ayant pour limite la taille disponible sur une clé USB : 5 Go.

Ayant seulement 7 objets à choisir, nous n'aurons que  $2^7$  combinaisons possibles (soit 128), ce qui est tout à fait raisonnable en terme de coût algorithmique. Nous allons créer tous les nombres binaires à 7 bits.

En voici deux exemples :

- Le nombre 1001100 signifie que nous avons choisi les fichiers 1, 4 et 5.
- Le nombre 1111111 signifie que nous avons choisi tous les fichiers.

1. Créer la liste de nombres binaires
2. Créer une fonction qui renverra une liste contenant tous les nombres binaires à  $n$  bits.
3. À partir de la liste de nombre binaires créées, créer une fonction qui renverra une liste de toutes les combinaisons de films possibles.

Recherche de la meilleure combinaison : Parmi toutes les combinaisons, il reste à choisir la meilleure, c'est à dire celle qui :

- maximise la durée d'écoute totale.
- ne dépasse pas la taille disponible sur la clé USB : 5 Go.

4. Créer une fonction qui renverra la meilleure solution parmi toutes celles disponibles dans une liste qui satisfont la taille limite. (Solution : la meilleure combinaison possible permet d'emporter 132 minutes de vidéos.)

## . APPLICATION : HARRY SE FAIT (ENFIN) LA MALLE !

### . DERNIERS PRÉPARATIFS AVANT LE DÉPART POUR POUDLARD

Harry a effectué tous ses achats. Maintenant qu'il est parfaitement équipé, il doit préparer sa malle avant le départ pour l'école de Poudlard.

Il dispose bien d'une malle pour ranger ses affaires mais cette malle est trop fragile pour supporter un poids important. Il ne pourra donc pas mettre toutes ses affaires dans la malle.

Voici la liste de fournitures scolaire des élèves de première année :

- 8 manuels scolaires :
  - Le Livre des sorts et enchantements (niveau 1), de Miranda Fauconnette
  - Histoire de la magie, de Bathilda Tourdesac
  - Magie théorique, de Adalbert Lasornette
  - Manuel de métamorphose à l'usage des débutants, de Emeric G. Changé
  - Milles herbes et champignons magiques, de Phyllida Augirolle
  - Potions magique, de Arsenius Beaulitron
  - Vie et habitat des animaux domestiques
  - Forces obscures : comment s'en protéger de Quentin Jentremble
- 1 baguette magique
- 1 chaudron (modèle standard en étain, taille 2)
- 1 boîte de fioles en verre ou cristal, varie également
- 1 télescope
- 1 balance en cuivre
- 3 robes de travail (noires)
- un chapeau pointu (noir)
- une paire de gants protecteurs (en cuir de dragon ou autre matière semblable)
- une cape d'hiver (noire avec attaches d'argent)

Un fichier est mis à votre disposition : les fournitures y sont disponibles dans une table de données. Chaque fourniture possède plusieurs caractéristiques (descripteurs) :

- son nom
- son poids (en kg)
- sa mana, c'est à dire son potentiel magique

Le fichier contient également le poids maximal supporté par la malle.

Remarque : pour simplifier ce projet, on ne considérera qu'un seul exemplaire par objet à emporter.

## . COMMENT REMPLIR CETTE MALLE ?

### **1. Remplir la malle... n'importe comment !**

Le premier objectif est de remplir la malle le plus vite possible, sans méthode particulière, peu importe ce qu'elle contient, tout en respectant le poids maximal qu'elle peut supporter.

Votre programme devra afficher le contenu de la malle et son poids.

### **2. Remplir la malle la plus lourde possible**

Vous devez maintenant concevoir un algorithme puis un programme qui remplira la malle en maximisant le poids des fournitures embarquées, sans toutefois dépasser le poids maximal supporté par la malle. Votre programme devra afficher le contenu de la malle et son poids.

### **3. Remplir la malle avec le maximum de mana (énergie magique)**

Vous devez maintenant concevoir un algorithme puis un programme qui remplira la malle en maximisant l'énergie magique (mana) des fournitures embarquées, sans toutefois dépasser le poids maximal supporté par la malle.

Votre programme devra afficher le contenu de la malle, sa valeur magique totale et son poids.

### **4. Améliorations possibles (pour les plus rapides)**

- Améliorer le remplissage de la malle en cherchant le meilleur rapport mana / poids des objets contenus dans la malle, afin d'obtenir un bon compromis.
- Ajouter un descripteur « Quantité » aux enregistrements de la table. Tenez compte du fait qu'il y a 8 manuels scolaires disponibles et 3 robes de travail. Adapter l'un ou autre des précédents programme à cette nouvelle contrainte.
- Ajouter une méthode par « force brute », qui apportera une solution optimale dans tous les cas (peu importe les objets à disposition).