
Les algorithmes gloutons - Exercices

QCM

1. Un algorithme glouton donne toujours :
 - ☐ la solution optimale à un problème
 - ☐ une solution en un temps relativement court
 - ☐ toutes les solutions
 - ☐ une solution en un temps relativement long
2. On remplit un camion de déménagement de 16 mètres cubes avec des palettes de volumes 10, 8, 5, 3, 1 mètres cubes. En raisonnant suivant un algorithme glouton, choisissez la combinaison que vous obtenez.
 - ☐ des palettes de 8 m³ , 5 m³ , 3 m³
 - ☐ des palettes de 10 m³ , 5 m³ , 1 m³
 - ☐ des palettes de 10 m³ , 3m³ , 3 m³
 - ☐ des palettes de 5 m³ , 5 m³ , 3 m³ , 3 m³
3. Qu'est-ce qu'un algorithme glouton ? (plusieurs réponses possibles)
 - ☐ C'est un algorithme qui suit le principe de faire, étape par étape, un choix optimum local.
 - ☐ C'est un algorithme qui tente de faire une optimisation globale.
 - ☐ C'est un algorithme qui optimise les temps de calcul.
 - ☐ C'est un algorithme qui prend la meilleure solution parmi toutes les solutions possibles.
4. Pour le jeu de pièces de valeurs (6,4,1) et pour une somme S=8 , choisir ce que renvoie l'algorithme glouton parmi les solution suivantes :
 - ☐ $8 = 6 + 1 + 1$
 - ☐ $8 = 4 + 4$
 - ☐ $8 = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$
 - ☐ rien

Exercice 1 (Problème du rendu de monnaie).

Un des grands classiques est le problème du rendu de monnaie où l'on souhaite rendre une somme en utilisant le moins de pièces (ou de billets) possibles. Le principe de l'algorithme glouton consiste à répéter le choix de la pièce de plus grande valeur, tant qu'elle ne dépasse pas la somme restant à rendre.

1. Complétez l'algorithme suivant en pseudo-code :

Algorithme : rendu_monnaie

/ Fournit la solution du rendu de monnaie à partir d'une liste de valeurs de chaque pièce ou billet, et de la somme à rendre */*

Entrées :

liste_valeur_monnaie : une liste des valeurs de chaque billet/pièce rangés par ordre décroissant

somme : la somme à rendre

Sorties :

choix : une liste de chaque billet/pièce à rendre

```
1 début
2   n ← longueur de la liste de pièces ;
3   choix ← n*[0] ;
4   pour i allant de 0 à n exclu par pas de 1 faire
5       tant que somme..... faire
6           somme ←.....;
7           choix[i] ←.....;
8       fin tq
9   fin pour
10  renvoyer choix
11 fin
```

2. Proposez une implémentation en Python. (l'exécution de rendu_monnaie([50, 20, 10, 5, 2, 1], 32) renvoie [0, 1, 1, 0, 1, 0])

Exercice 2 (Caisse d'un grand magasin).

Vous travaillez à la caisse d'un grand magasin parisien, à la veille des fêtes de Noël. Vous devez rendre 99 € au premier client, et 19 € au deuxième client. Vous disposez en caisse des pièces et billets suivants (en €) : [50, 20, 10, 5, 2, 1]. La caisse ne contient que deux exemplaires de chaque.

1. En utilisant un algorithme glouton, indiquez quelles pièces et billets vous devez rendre au premier client et le nombre de pièces et billets restant en caisse.
2. En utilisant un algorithme glouton, indiquez si vous pouvez rendre la monnaie au deuxième client.
3. On souhaite procéder à une implémentation Python de cet algorithme glouton. Pour cela, on définit deux listes :
 - *liste_valeur_monnaie* qui prend la valeur [50, 20, 10, 5, 2, 1] (la valeur faciale des billets et pièces en euros)
 - *caisse* qui prend la valeur [2,2,2,2,2,2] (le nombre de chaque billet et pièce dans la caisse).

Complétez la fonction suivante (on pourra utiliser les réponses précédentes comme exemple d'exécution) :

```
1 def magasin(somme, liste_valeurs_monnaie, caisse):
2     n = len(liste_valeurs_monnaie)
3     choix = n*[0]
4     for i in range (0, n, 1):
5         while somme >= liste_valeurs_monnaie[i] and ..... :
6             somme = .....
7             choix[i] = .....
8             caisse[i] = .....
9     return (choix, caisse)
```

Exercice 3 (Range ta chambre !).

Votre chambre est mal rangée (une fois n'est pas coutume). Votre mère vous demande de ranger les livres d'informatique et les romans qui traînent sur le sol. Pour cela, elle vous achète une bibliothèque, dont les étagères sont de longueur L fixées. Il y en a 5. Chaque livre possède un titre et une largeur. La collection de livre est indiquée par deux listes : la première contient le titre des livres, la deuxième contient la largeur des livres. Chacune de ces listes est triée de telle façon que :

- `titre[i]` correspond à `largeur[i]`,
- chaque liste est triée par largeur décroissante.

1. Pour minimiser le nombre d'étagères, on se propose d'écrire une fonction *rangement* prenant *largeur*, *titre* et L en argument, et renvoyant une liste de tous les livres rangés sur une seule étagère, ainsi que les deux listes de livre à restant à classer (titre et largeur). Utilisez un algorithme glouton pour écrire cette fonction.

Vous pourrez utiliser les jeux de données suivants :

```
1  L = 1.00
2  titre = ["Les frères Karamazov",
3           "Nana",
4           "Germinal",
5           "Le rouge et le noir",
6           "Le temps est assassin",
7           "Pot-Bouille",
8           "Cadres Noirs",
9           "Antigone",
10          "Le petit prince",
11          "La ferme des animaux",
12          "1984",
13          "Apprenez à programmer en python",
14          "La fabrique des objets connectés",
15          "Tout ce qui est sur terre doit périr",
16          "Linux pour les nuls"]
17  largeur = [0.55,
18             0.37,
19             0.25,
20             0.22,
21             0.22,
22             0.21,
23             0.18,
24             0.17,
25             0.14,
26             0.14,
27             0.13,
28             0.10,
29             0.09,
30             0.08,
31             0.07]
```

2. Proposez une fonction *tout_ranger* renvoyant le contenu de chaque étagère.

Exercice 4 (Gestion d'un cinéma).

Un cinéma possède S salles de cinéma. Chaque semaine, le cinéma propose une liste de films à voir. Chaque film correspond à un nombre fini de séance.

Chaque séance est caractérisée par un tuple (**titre**, **debut**, **fin**), dans lequel :

- **titre** désigne le titre du film,
- **debut** désigne l'heure de début du film,
- **fin** désigne l'heure de fin du film.

Un étudiant veut voir le maximum de films pendant la première journée du « Printemps du Cinéma ».

Ce problème peut être résolu par un algorithme glouton.

1. Première possibilité : l'étudiant choisi les films en fonction de l'heure de fin. Proposez une implémentation Python de cet algorithme glouton. ^a
2. Deuxième possibilité : l'étudiant choisi les films en fonction de leur durée. Proposez un algorithme glouton. ^b

On pourra tester avec la liste des films ci-dessous ^c :

```
1 films = [("Cyrille, agriculteur, 30 ans", 14.25, 16.25),
2         ("Cyrille, agriculteur, 30 ans", 16.50, 18.50),
3         ("Invisible Man", 10.50, 12.75),
4         ("Invisible Man", 18.50, 20.75),
5         ("Invisible Man", 21.00, 22.25),
6         ("Judy", 16.00, 18.25),
7         ("Judy", 21.00, 23.25),
8         ("Mine de rien", 14.25, 15.75),
9         ("Mine de rien", 18.75, 20.25),
10        ("Mine de rien", 20.50, 22.00),
11        ("10 jours sans maman", 14.00, 15.75),
12        ("10 jours sans maman", 16.25, 18.00),
13        ("10 jours sans maman", 18.30, 20.25),
14        ("L'appel de la forêt", 10.50, 12.25),
15        ("L'appel de la forêt", 14.25, 16.00),
16        ("L'appel de la forêt", 16.50, 18.25),
17        ("L'appel de la forêt", 18.75, 20.50),
18        ("Le cas Richard Jewell", 20.75, 23.00),
19        ("Le prince oublié", 16.25, 17.00),
20        ("Sonic le film", 10.50, 12.25),
21        ("Sonic le film", 14.25, 16.00),
22        ("Sonic le film", 16.50, 18.25),
23        ("#Jesuisla", 20.75, 22.50),
24        ("Birds of Prey", 21.00, 22.75),
25        ("Un soir en Toscane", 18.50, 20.25),
26        ("Au nom de la terre", 18.75, 20.50)
27    ]
28
```

a. L'étudiant choisit le film qui finit le plus tôt, puis celui d'après à condition que son heure de début soit compatible et qu'il ne l'ait pas déjà vu. En cas d'égalité, proposez une solution.

b. L'étudiant choisit le film qui dure le moins longtemps, puis celui d'après à condition que son heure de début soit compatible avec l'heure de fin du précédent

c. Horaires du cinéma "Les Arches Lumière" d'Yvetot du dimanche 1^{er} mars 2020