

Concepts algorithmiques

Rappel:

Un algorithme est une suite d'instructions pour réaliser une tâche ou résoudre un problème.

Il existe souvent plusieurs algorithmes pour résoudre un problème. On peut alors comparer leur performance en termes d'espace mémoire ou de temps nécessaire. Un algorithme brutal (recherche systématique de toutes les solutions possibles sans stratégie) n'est pas performant car il nécessite beaucoup de mémoire et de temps pour se finaliser.

Définition:

Dans un algorithme glouton, on procède étape par étape en faisant, à chaque étape, le meilleur choix possible. on ne remet jamais en question les choix faits aux étapes précédentes.

Activité:

Voici notre système de monnaie (exprimé en euros) : Pièces : 0,01 ; 0,02 ; 0,05 ; 0,10 ; 0,20 ; 1 ; 2.

Billets : 5 ; 10 ; 20 ; 50 ; 100 ; 200 ; 500.

On cherche à totaliser 53 euros **avec un nombre minimal de pièces et billets**.

a. Avec une stratégie gloutonne, quelle valeur sera choisie en premier ? Quelle sera la solution ?

b. Voici 2 solutions possibles affichées dans le tableau ci-dessous.

Utiliser l'approche brutale pour proposer 2 autres solutions.

Écrire la solution optimale à la dernière ligne :

Solutions	Nombre de pièces et de billets
5300 x 0,01 €	5300
53 x 1 €	53

c. Est-ce que la stratégie gloutonne offre la solution optimale dans ce cas ?

d. Utiliser l'algorithme glouton pour totaliser 14 € **avec un nombre minimal de pièces et billets**.

Voici le système de monnaie imaginaire $S = [1, 2, 5, 7, 10, 20]$.

L'algorithme glouton donne-t-il la solution optimale dans ce cas ? Sinon, quelle est la solution optimale ?

En résumé, nous avons vu:

- Pour résoudre un problème, il existe souvent plusieurs algorithmes qui peuvent être plus ou moins efficaces.
- L'approche brutale consiste à tester toutes les solutions possibles.
- La stratégie gloutonne est souvent employée pour des problèmes d'optimisation. Cette stratégie ne donne pas toujours la solution optimale.

Exercices :

Ex.1 Utiliser la stratégie gloutonne pour le rendu de monnaie optimal d'une somme de 27 € dans le cas d'un système $S = [1, 2, 5, 100]$.

Ex.2 Avant la décimalisation définitive de 1971, le système monétaire britannique repose sur une subdivision de valeur qui remontait à l'époque carolingienne. Voici les principales valeurs de cet ancien système, appelé **système impérial** :

- 1 penny
- 3 pence (= pluriel de penny)
- 6 pence (= demi-Shilling)
- 12 pence (= Shilling)
- 24 pence (= Florin)
- 30 pence (= demi-couronne)
- 60 pence (= Couronne / Crown)
- 240 pence (= Livre / Pound)
- 960 pence (= Farthing)



- Quel sera le choix de monnaies optimal pour totaliser 48 pences avec le système impérial ($S = [1, 3, 6, 12, 24, 30, 60, 240, 960]$) dans le cas où on utilise l'algorithme glouton ?

- Y a-t-il une solution plus optimale ?

Ex.3 Comment gérer le rendu de monnaie optimal d'une somme de 27 € dans le cas d'un système $S = [1, 2, 5, 50, 100]$, sachant qu'on ne dispose que quatre pièces de chaque type ?