

Rendu de monnaie optimal

Tor’Khan Restricted - Gabriel ABGRALL

2024-11-16

Table des matières

Rendu de monnaie optimal	2
1. Définition du projet	2
1.1 Systèmes monétaires utilisés	2
2. Algorithmes	3
2.1. Algorithme glouton	3
A vos claviers !	3
2.2. Recherche d'un algorithme optimal	3
A vos claviers !	4
3. Pour aller plus loin	5
3.1. Généralisation de l'algorithme	5
A vos claviers !	5
3.2. Optimisation de l'algorithme	5
A vos claviers !	5
3.3. Comparaison	5
A vos claviers !	6

Rendu de monnaie optimal

1. Définition du projet

L'objectif du projet est de créer un algorithme capable de rendre la monnaie pour n'importe quelle somme donnée, en utilisant n'importe quel système monétaire, et en utilisant le moins de pièces possibles.

1.1 Systèmes monétaires utilisés

Dans le cadre du projet, nous utiliserons ces deux systèmes comme exemple, contenant les pièces suivantes :

- Système euro : {50, 20, 10, 5, 2, 1}
- Système impérial : {30, 24, 12, 6, 3, 1}

2. Algorithmes

2.1. Algorithme glouton

Dans un premier temps, nous pouvons créer un algorithme glouton.

Son objectif est de prendre la valeur la plus grande autant de fois que possible, puis diminuer progressivement jusqu'à arriver à la somme demandée.

```
1 Exemple :  
2  
3 Nous devons rendre 48 euros.  
4  
5 Pour le système euro, l'algorithme choisit donc (dans l'ordre) :  
6 - 0 pièce de 50  
7 - 2 pièce de 20  
8 - 0 pièce de 10  
9 - 1 pièce de 5  
10 - 1 pièce de 2  
11 - 1 pièce de 1  
12  
13 Pour le système impérial, l'algorithme choisit :  
14 - 1 pièce de 30  
15 - 0 pièce de 24  
16 - 1 pièce de 12  
17 - 1 pièce de 6
```

A vos claviers !

Ecrivez un algorithme glouton et résolvez l'exercice ci-dessus pour n'importe quelle somme donnée.

2.2. Recherche d'un algorithme optimal

Nous constatons alors que l'algorithme glouton n'est pas optimal pour tous les systèmes. Bien qu'il ait permis de trouver la solution pour le système européen, il n'a pas été capable de la trouver pour le système impérial. En effet, pour l'exemple ci-dessus, nous aurions dû rendre deux pièces de 24, ce qui représente une pièce de moins que la solution trouvée par l'algorithme.

Il faut donc inventer un algorithme capable de trouver la solution optimale pour le système impérial, et pour n'importe quelle somme donnée.

A vos claviers !

Trouvez et écrivez un algorithme permettant de rendre la monnaie de façon optimale (le moins de pièces possible) en utilisant le système impérial.

Validez l'étape 2 complètement avant de passer à l'étape 3.

3. Pour aller plus loin

3.1. Généralisation de l'algorithme

Maintenant que nous avons un algorithme optimal pour le système impérial, ci ce n'est pas déjà le cas, nous souhaitons un être capables de résoudre ce problème pour n'importe quel système monétaire.

A vos claviers !

Généralisez votre algorithme (ou trouvez en un autre) pour trouver la solution optimale pour n'importe quel système de monnaie et pour n'importe quelle somme initiale.

3.2. Optimisation de l'algorithme

Votre algorithme permet de trouver le rendu optimal pour n'importe quelle somme et pour n'importe quel système monétaire ? C'est bien. Maintenant, optimisez cet algorithme pour pouvoir résoudre des sommes astronomiques en utilisant de la petite monnaie.

A vos claviers !

Optimisez votre algorithme pour trouver la solution à toutes les sommes à partir de 1 jusqu'au plus grand nombre possible en moins de 5 secondes.

Variante : Optimisez votre algorithme pour trouver une somme unique la plus élevée possible en moins de 5 secondes

3.3. Comparaison

Nous pouvons maintenant trouver toutes les combinaisons optimales pour chaque somme et pour chaque système monétaire donné. Nous voulons maintenant comparer ces résultats avec l'algorithme glouton, en effectuant un pourcentage des réponses correctes qu'il permet de trouver.

Nous souhaitons aussi afficher chaque réponse non optimales que l'algorithme glouton obtient, et la réponse optimale trouvée par notre algorithme.

A vos claviers !

Améliorer votre programme en affichant chaque solution optimale trouvée que l'algorithme glouton ne permettait pas de calculer pour toutes les sommes entre 1 et un nombre choisis. A la fin du calcul, afficher le temps de calcul total, ainsi que le pourcentage de réponses correctes de l'algorithme glouton.