Algorithme glouton
Problème du rendu de monnaie

Christophe Viroulaud

Première - NSI

Algo 10

Algorithme glouton
Problème du rendu de monnaie

Approche exhaustive

Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Figure 1 – Caisse automatique dans une boulangerie

Mettre en œuvre une nouvelle technique algorithmique.

Algorithme glouton
Problème du rendu de monnaie

Approche exhaustive

Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Problématique

Rendre la monnaie en donnant le moins de pièces possibles.

Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Conditions initiales

- Nous limiterons le problème aux valeurs entières.
- Nous n'utiliserons que les billets ou pièces de 1, 2, 5, 10€.
- On supposera que la caisse contient une quantité illimitée de monnaie.

Sommaire

1. Approche exhaustive

2. Approche gloutonne

Approche exhaustive

Pour choisir la *meilleure* combinaison, une solution consiste à tester toutes les combinaisons possibles avec les pièces dont nous disposons.

Algorithme glouton
Problème du rendu de monnaie

Approche exhaustive

Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

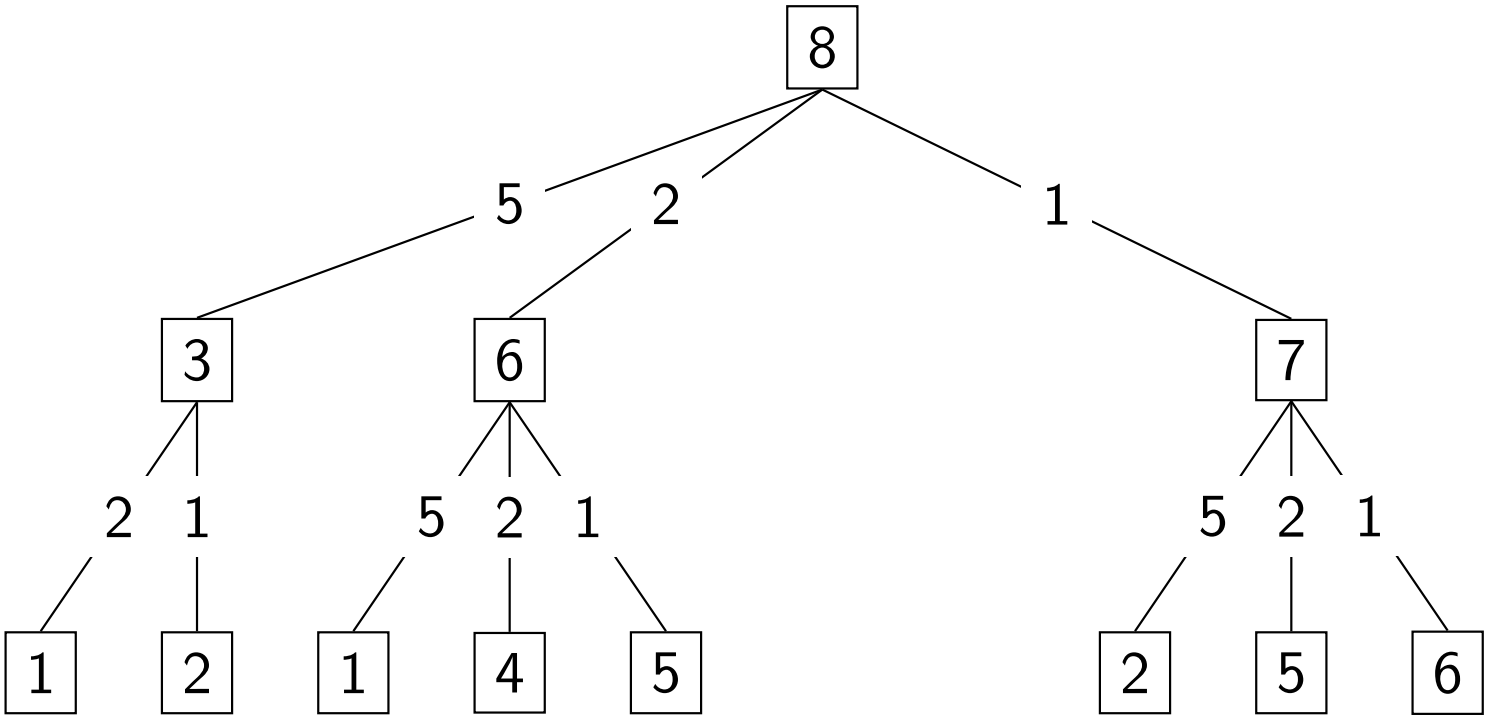

Figure 2 – Construire les combinaisons pour rendre 8€

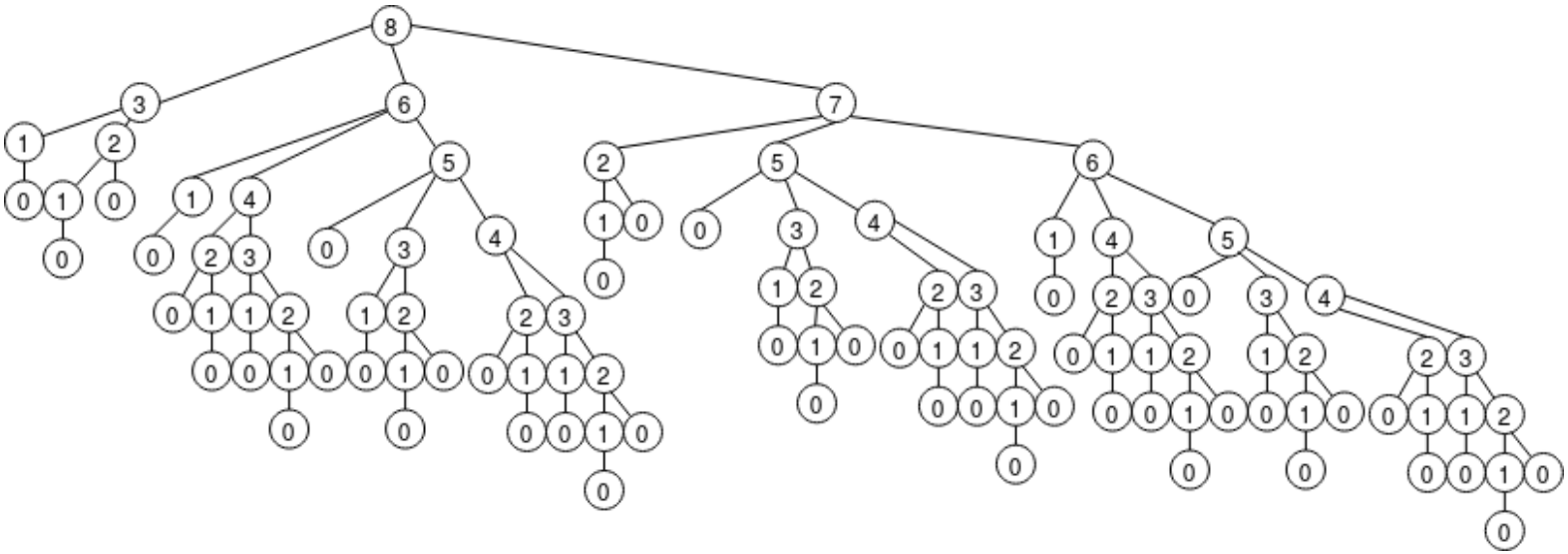

Figure 3 – Rendre 8€ : arbre complet des possibilités

Algorithme glouton
Problème du rendu de monnaie

Approche exhaustive

Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

À retenir

Certains problèmes d'apparence simple cachent un coût temporel de résolution important. Le nombre de solutions devient rapidement très grand pour des cas mêmes simples : on parle **d'explosion combinatoire**.

Sommaire

1. Approche exhaustive

2. Approche gloutonne
2.1 Principe
2.2 Propriétés
2.3 Implémentation
2.4 Efficacité

Approche gloutonne - principe

À retenir

Pour éviter d'énumérer toutes les solutions, l'approche gloutonne consiste à effectuer un seul choix à chaque étape et ne pas revenir dessus.

À retenir

Un algorithme glouton construit un résultat global par une succession de choix intermédiaire donnant systématiquement le meilleur résultat partiel.

Approche gloutonne

Pour le rendu de monnaie, le choix est fait de **rendre la plus grande pièce possible à chaque étape.**

Figure 4 – Approche gloutonne du rendu de monnaie : rendre la plus grande pièce possible à chaque étape.

Sommaire

1. Approche exhaustive

2. Approche gloutonne
2.1 Principe
2.2 Propriétés
2.3 Implémentation
2.4 Efficacité

Propriétés

À retenir

Une approche gloutonne est :

- plus rapide qu'une solution exhaustive,

Propriétés

À retenir

Une approche gloutonne est :

- plus rapide qu'une solution exhaustive,
- plus simple à implémenter,

Propriétés

À retenir

Une approche gloutonne est :

- plus rapide qu'une solution exhaustive,
- plus simple à implémenter,
- **ne donne pas forcément la meilleure solution**

Algorithme glouton
Problème du rendu de monnaie
Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Activité 1 : L'ancien système monétaire britannique était composé des pièces : 1, 3, 4.

1. En appliquant l'algorithme glouton, donner les pièces utilisées pour rendre 6£.
2. Existe-t-il une meilleure solution ?

Correction

Pour rendre 6 :

- avec l'approche gloutonne : 4, 1, 1
- solution optimale : 3, 3

Algorithme glouton
Problème du rendu de monnaie
Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Sommaire

1. Approche exhaustive

2. Approche gloutonne
2.1 Principe
2.2 Propriétés
2.3 Implémentation
2.4 Efficacité

Implémentation

```
1 systeme = [10, 5, 2, 1]
```

Code 1 – On représente le système monétaire par un tableau.

Activité 2 : Écrire la fonction **`rendu(somme: int, systeme: list)`** $\rightarrow$ **`list`** qui renvoie le tableau des pièces à choisir pour rendre **`somme`**. La fonction implémentera l'algorithme suivant :
Tant que la somme à rendre n'est pas nulle :

- Si la pièce du système est strictement supérieure à la somme :
  - Avancer dans le système monétaire
- Sinon :
  - Stocker la pièce,
  - Mettre à jour la somme

Algorithme glouton
Problème du rendu de monnaie
Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité

Aide

Aide

Pour *avancer* dans le système monétaire, il faut initialiser un indice à 0 puis l'incrémenter si la pièce proposée est trop grande.

Correction

```python
def rendu_monnaie(somme: int, systeme: list) -> list:
    res = []
    i_piece = 0
    while somme > 0:
        # si pièce est trop grande
        if systeme[i_piece] > somme:
            # on avance dans le système
            i_piece = i_piece + 1
        else:
            res.append(systeme[i_piece])
            somme = somme - systeme[i_piece]
    return res

# appel de la fonction
assert rendu_monnaie(14, [10, 5, 2, 1]) == [10, 2, 2]
```

Sommaire

1. Approche exhaustive

2. Approche gloutonne
2.1 Principe
2.2 Propriétés
2.3 Implémentation
2.4 Efficacité

Efficacité

Dans le **pire des cas**, on ne rend que des pièces de 1€. Dans ce cas la boucle `while` effectuera autant de tours que la valeur de `somme`.

À retenir

On peut dire que, dans le pire des cas, l'approche gloutonne est linéaire par rapport à la somme à rendre.

Algorithme glouton
Problème du rendu de monnaie
Approche exhaustive
Approche gloutonne
Principe
Propriétés
Implémentation
Efficacité