Algorithmes Avancés

1. Introduction aux Algorithmes Avancés

Les **algorithmes avancés** regroupent un ensemble de méthodes et de stratégies permettant de résoudre des problèmes complexes de manière efficace. Ils deviennent indispensables lorsque les approches classiques (boucles simples, conditions, recherche exhaustive, force brute) atteignent leurs limites en termes de :

- **Complexité temporelle**: temps nécessaire à l'exécution de l'algorithme.
- **Complexité spatiale**: quantité de mémoire utilisée par l'algorithme.

La maîtrise de ces techniques constitue une étape importante dans la formation d'un développeur, car elle permet de concevoir des programmes plus performants, plus élégants et capables de traiter de grands volumes de données.

Les principaux paradigmes étudiés dans ce chapitre sont :

- Diviser pour Régner (*Divide and Conquer*) ;
- La Programmation Dynamique (*Dynamic Programming*) ;
- Les Algorithmes Gloutons (*Greedy Algorithms*).

2. Le Paradigme « Diviser pour Régner » (Divide and Conquer)

L'approche **Diviser pour Régner** consiste à décomposer un problème complexe en plusieurs sous-problèmes plus simples, à résoudre chacun d'eux indépendamment, puis à combiner leurs solutions afin d'obtenir la solution finale.

Cette méthode se déroule en trois étapes :

1. **Diviser :** découper le problème initial en plusieurs sous-problèmes de taille réduite.
2. **Régner :** résoudre chaque sous-problème de manière récursive.
3. **Combiner :** fusionner les résultats obtenus pour construire la solution globale.

Exemples d'application :

- **Tri Fusion (Merge Sort)**
- **Tri Rapide (Quick Sort)**

apcpedagogie.com

Ces algorithmes possèdent généralement une complexité de **O(n log n)**, ce qui les rend beaucoup plus performants que les méthodes de tri quadratiques en **O(n²)**.

3. La Programmation Dynamique (Dynamic Programming)

La **programmation dynamique** est une technique d'optimisation utilisée lorsque le problème étudié peut être décomposé en plusieurs sous-problèmes qui se répètent.

L'idée principale consiste à éviter de recalculer plusieurs fois les mêmes résultats en les mémorisant dans une structure de données.

Techniques principales

- **Mémoïsation (Top-Down)**: approche récursive dans laquelle les résultats déjà calculés sont stockés dans un tableau ou un dictionnaire.
- **Tabulation (Bottom-Up)**: approche itérative qui construit progressivement la solution à partir des cas les plus simples.

Exemples d'application :

- Calcul optimisé de la suite de Fibonacci ;
- Problème du Sac à Dos (*Knapsack Problem*) ;
- Calcul du plus long sous-tableau commun ;
- Problèmes de planification et d'optimisation.

4. Les Algorithmes Gloutons (Greedy Algorithms)

Un **algorithme glouton** construit progressivement une solution en effectuant, à chaque étape, le choix qui paraît être le meilleur localement, sans revenir sur les décisions déjà prises.

Avantages

- Rapides et faciles à implémenter ;
- Faible consommation de mémoire ;
- Très efficaces pour certains problèmes d'optimisation.

apcpedagogie.com

Inconvénients

- Ne garantissent pas toujours la solution optimale ;
- Leur efficacité dépend fortement de la nature du problème traité.

Exemples d'application :

- Algorithme de Dijkstra ;
- Codage de Huffman ;
- Problème du rendu de monnaie.

Pseudo-code : Rendu de Monnaie

```
Fonction RenduMonnaie(Montant : Entier,
                      PiecesDisponibles : Tableau d'Entiers) : Liste

Variables
    Resultat : Liste

Début
    Resultat ← ListeVide

    Pour chaque Piece dans PiecesDisponibles Faire
        TantQue Montant ≥ Piece Faire
            Ajouter Piece à Resultat
            Montant ← Montant - Piece
        FinTantQue
    FinPour

    Retourner Resultat
Fin
```

5. Simulateur Interactif : Algorithme Glouton

Ce simulateur permet de visualiser le fonctionnement de l'algorithme glouton appliqué au problème du rendu de monnaie. L'utilisateur saisit un montant, puis observe étape par étape le choix effectué par l'algorithme.

Pièces disponibles : 50, 20, 10, 5, 2 et 1.

apcpedagogie.com

Pour chaque étape, le simulateur affiche :

- Le montant restant ;
- La pièce choisie ;
- Le nouveau reste ;
- Le nombre total de pièces utilisées.

Rédigé et structuré par : **Med Khalil Kribi**

Source : https://apcpedagogie.com/algorithmes-avances · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com