

TD n°9

Revision

Exercice 1 Récursion

Donner les bornes asymptotiques pour les fonctions suivantes sachant que pour $n \leq 2$ elles sont constantes :

- { $T(n) = 7T(n/2) + n^2$
- { $T(n) = 3T(n/2) + n \log n$
- { $T(n) = 4T(n/2) + n^2\sqrt{n}$

Exercice 2 Pièces de monnaies

On considère le problème où l'on doit rendre la monnaie pour x euros avec le moins de pièces de monnaies possible.

1. On a des pièces de 1, 2, 5, 10. Écrire l'algorithme glouton qui donne une solution optimale.
2. On dispose d'un ensemble de pièces $c^0, c^1, c^2, \dots, c^k$ pour $c > 1$, et $k \geq 1$. Écrire l'algorithme glouton qui donne la solution optimale.
3. Donner un ensemble de pièces tel que l'algorithme glouton ne retourne pas une solution optimale.
4. Donner un ensemble de pièces tel que l'algorithme glouton ne trouve pas de solution.
5. Donner un algorithme qui retourne une solution optimale pour n'importe quel ensemble de pièces.

Exercice 3 L'élément majoritaire

Soit E un ensemble de n éléments rangés dans un tableau numéroté de 1 à n . On suppose que la seule opération qu'on sait effectuer sur les éléments est de vérifier si deux éléments sont égaux ou non. On dit qu'un élément $x \in E$ est majoritaire si l'ensemble $E_x = \{y \in E | y = x\}$ a strictement plus de $n/2$ éléments. On supposera que n est une puissance de 2.

1. Écrire un algorithme calculant le cardinal c_x de E_x pour un x donné. En déduire un algorithme pour vérifier si E possède un élément majoritaire. Quelle est la complexité de cet algorithme?
2. Donner un autre algorithme récursif basé sur un découpage de E en deux listes de même taille. Quelle est sa complexité?
3. Pour améliorer l'algorithme précédent, on va se contenter dans un premier temps de mettre au point un algorithme possédant la propriété suivante :
 - { soit l'algorithme garantit que E ne possède pas d'élément majoritaire,
 - { soit l'algorithme fournit un entier $p > n/2$ et un élément x tels que x apparaisse au plus p fois dans E et tout élément autre que x apparaisse au plus $n - p$ fois dans E .Donner un algorithme possédant cette propriété. Quelle est sa complexité?
4. En déduire un algorithme efficace vérifiant si E possède un élément majoritaire.