

TD d'algorithmique avancée

TD 2 : récursivité

Jean-Michel Dischler et Frédéric Vivien

Suite de Fibonacci

La suite de Fibonacci est définie comme suit :

$$\text{Fib}(n) = \begin{cases} 1 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & \text{sinon.} \end{cases}$$

1. Écrivez un algorithme récursif calculant $\text{Fib}(n)$.
2. Montrez que la complexité (en nombre d'additions) de cet algorithme est en $\Omega(2^{\frac{n}{2}})$.
3. Écrire un algorithme récursif qui calcule, pour $n > 0$, le couple $(\text{FIBONACCI}(n), \text{FIBONACCI}(n-1))$.
4. Utilisez l'algorithme précédent pour écrire un nouvel algorithme calculant $\text{FIBONACCI}(n)$.
5. Qu'elle est la complexité (en nombre d'additions) de cet algorithme ?

Opérations ensemblistes

Dans cette partie on considère des ensembles représentés par des tableaux, certains ensembles seront triés et d'autres pas. **Toutes les solutions proposées doivent être récursives.**

1. Nous voulons un algorithme $\text{APPARTENANCE}(A, x)$ qui recherche si un élément x appartient à l'ensemble A . Si x appartient effectivement à A , l'algorithme renverra VRAI, et FAUX sinon.
 - (a) Cas des ensembles non triés :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité en nombre de comparaisons ?
 - (b) Cas des ensembles triés (dans l'ordre croissant) :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité en nombre de comparaisons ?
 - iii. Utilisez une recherche dichotomique pour améliorer votre algorithme.
 - iv. Quelle est la complexité de votre nouvel algorithme ?
2. Nous voulons maintenant un algorithme $\text{UNION}(A, B)$ qui nous renvoie l'union des deux ensembles qui lui sont passés en argument.
 - (a) Cas des ensembles non triés :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité ?
 - (b) Cas des ensembles triés (dans l'ordre croissant) :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité ?
3. Nous voulons maintenant un algorithme $\text{INTERSECTION}(A, B)$ qui nous renvoie l'intersection des deux ensembles qui lui sont passés en argument.

- (a) Cas des ensembles non triés :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité?
 - (b) Cas des ensembles triés (dans l'ordre croissant) :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité?
4. Nous voulons maintenant un algorithme DIFFÉRENCE(A, B) qui nous renvoie la différence des deux ensembles qui lui sont passés en argument (La différence de A et de B , notée $A \setminus B$ est l'ensemble des éléments de A n'appartenant pas à B).
- (a) Cas des ensembles non triés :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité?
 - (b) Cas des ensembles triés (dans l'ordre croissant) :
 - i. Écrivez un tel algorithme.
 - ii. Quelle est sa complexité?