

Techniques mathématiques pour l'informatique
TD – Séance n° 4

Exercice 1 – On veut tester récursivement si deux tableaux $T1$ et $T2$ de n entiers sont égaux : pour cela, on se base sur la propriété que deux tableaux sont égaux s'ils ont le même premier élément et si les deux tableaux privés de leur premier élément sont égaux.

Question 1 : Ecrire la fonction $tabEgaux(T1, T2, i)$ qui renvoie vrai si les i derniers éléments de $T1$ et $T2$ sont égaux et renvoie faux sinon (pour résoudre notre problème il suffira d'appeler $tabEgaux(T1, T2, n)$).

Question 2 : Evaluer la complexité de cet algorithme.

On veut tester récursivement si un tableau $T1$ de taille n_1 est contenu dans un tableau $T2$ de taille n_2 . Pour cela, on se base sur le fait que $T1$ est contenu dans $T2$ si son premier élément est contenu dans $T2$ et si $T1$ privé de son premier élément est contenu dans $T2$.

Question 3 : Ecrire la fonction $tabContenu(T1, T2, i)$ qui renvoie vrai si les i derniers éléments de $T1$ sont contenus dans $T2$, et renvoie faux sinon (pour résoudre notre problème il suffira d'appeler $tabContenu(T1, T2, n)$).

Question 4 : Evaluer la complexité de cet algorithme.

Exercice 2 – Associer les fonctions de gauches avec les classes de droites selon leurs appartenances :

$\log(\log(n))$	•	
$\frac{6}{8}n + n$	•	
$2\sqrt{n}$	•	• $\theta(n^3)$
$n(n-3)(n+3)$	•	• $O(n^2)$
$\frac{\log(n)}{n}$	•	• $O(4^n)$
$2^6n - 4^3n + 1$	•	• $O(n \log(n))$
$\frac{(n-1)(n+1)\log(n)}{6n}$	•	
$0, 1n^3 + n^2$	•	