

Algorithmique et Complexité
Examen

Seules les notes manuscrites de cours et de TD sont autorisées.

Exercice 1 – Complexité

Parmi les affirmations suivantes, lesquelles sont vraies ? a et b sont des entiers positifs.

$f(n)$	$g(n)$	$f = O(g)$	$g = O(f)$
$n + 1$	$n^4 + 3n - 2$		
$n + 4$	$4n$		
$\frac{1}{10}n^3 + 100n^2 + 100000$	n^3		
2^{n+1}	2^n		
2^{2^n}	2^n		
$(n + a)^b$	n^b		
n^n	$n!$		

Exercice 2 – Arbres binaires de recherche

On suppose que les entiers compris entre 1 et 1000 sont disposés dans un arbre binaire de recherche, et on souhaite retrouver le nombre 363. Parmi les séquences suivantes, lesquelles pourraient et ne pourraient pas être la séquence de nœuds parcourus ? Vous justifierez votre réponse.

1. 2, 252, 401, 398, 330, 344, 397, 363.
2. 924, 220, 911, 244, 898, 258, 362, 363.
3. 925, 202, 911, 240, 912, 245, 363.
4. 2, 399, 387, 219, 266, 382, 381, 278, 363.
5. 935, 278, 347, 621, 299, 392, 358, 363.

Exercice 3 – Les Arbres

Soit la structure de données suivante :

```
Enregistrement Nœud {  
    Val      : entier;  
    Gauche  : ↑ Nœud;  
    Droit   : ↑ Nœud;  
}
```

1. Ecrire un algorithme qui compte le nombre de feuilles d'un arbre binaire
2. Ecrire un algorithme qui compte le nombre de nœuds d'un arbre binaire ayant deux fils avec l'un des deux fils (et pas les deux) étant une feuille.
3. Ecrire un algorithme qui calcule la hauteur d'un arbre.
4. Ecrire un algorithme qui étant donné un arbre binaire contenant uniquement des valeurs entières renvoie vrai si l'arbre est un arbre binaire de recherche et faux sinon.

Exercice 4 – Rendu de monnaie

Pour simplifier, les tableaux seront indicés à partir de 1 dans cet exercice. Un tableau représente le contenu d'une caisse enregistreuse. Chaque case contient la valeur d'une pièce de cette caisse. On suppose que la plus grande valeur contenue dans la caisse est bornée par 10.

1. Proposer un algorithme qui prend en entrée un tableau T et calcule combien d'éléments de chaque valeur se trouve dans T . La sortie de votre algorithme sera un tableau Occurrences de 10 cases dont la i -ème case contient le nombre de fois que i apparaît dans T .
2. Donner la complexité de votre algorithme.
3. Proposer un algorithme de rendu de monnaie qui prend en entrée le tableau Occurrences calculé à la question précédente et une valeur x de monnaie à rendre. L'algorithme renvoie un tableau Rendu de 10 cases qui représente la monnaie rendue pour atteindre x . La i -ème case de Rendu contient le nombre de fois que la pièce de valeur i a été rendue.

Par exemple si vous avez huit pièces de 1, deux pièces de 2 et une pièce de 5 et que vous voulez rendre 11 vous pouvez rendre une pièce de 5, deux pièces de 2 et deux pièces de 1.

Indice : algorithme glouton.

4. Donner la complexité de votre algorithme.