U.S.T.H.B - 2025-2026
Faculté d'Informatique
N. Bensaou

Master Informatique Visuelle
Algorithmique et Complexité (M1 - MIV)

Série 3 : Les équations de récurrence

Exercice 1.- Résoudre les équations de récurrences suivantes :

a. $g(n) = \begin{cases} 0 & \text{si} \quad n = 0 \\ g(n-1) + 2 * n - 1 & \text{si} \quad n > 0 \end{cases}$

b. $T(m,n) = \begin{cases} 2 * T(m/2, n/2) + m * n & \text{si} \quad m > 1, n > 1, m <= n \\ n & \text{si} \quad m = 1 \\ m & \text{si} \quad n = 1 \\ n + m & \text{si} \quad n = 1 \text{si} \quad m = 1 \end{cases}$

c. $T(n) = \begin{cases} 1 & \text{si} \quad n = 0 \\ T(n-1) + 2^n & \text{si} \quad n > 0 \end{cases}$

d. $T(n) = \begin{cases} 1 & \text{si} \quad n = 1 \\ 7 * T(n/2) & \text{si} \quad n > 1 \end{cases}$

Exercice 2.- Soit l'équation de récurrence :

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ \sum_{i=1}^{i=n-1} T(i) + 1 & \text{si } n \geq 2 \end{cases}$$

a.) Calculez $T(n) - T(n-1)$, pour $n \geq 2$.

b.) Résoudre $T(n)$.

Exercice 3.- Prouvez par induction sur n que $\sum_{i=0}^{i=n} F_i = F_{n+2} - 1$ où $F_i, i = 1, 2, \ldots, n, \ldots$ sont les nombres de Fibonacci.

Rappelons que les nombres de Fibonacci sont définis par la récurrence suivante :

$F_n = \begin{cases} 1 & \text{si} n = 0, 1 \\ F_{n-1} + F_{n-2} & \text{si} n \geq 2 \end{cases}$

Exercice 4.- Soit $S[1..n]$ un tableau d'entiers et x un entier donné.

a- Écrire un algorithme naïf qui permet de vérifier s'il existe ou non dans S deux entiers x_1, x_2 tels que $x = x_1 + x_2$.

b- Donnez sa complexité.

c- Décrire un algorithme en $\Theta(n \log n)$ qui résout ce problème. Justifiez correctement votre réponse.

Exercice 5.- Soit $A[1..n]$ un tableau de n entiers donnés triés en ordre croissant.

a- Écrire une algorithme qui détermine l'élément qui se répète le plus dans A ainsi que sa fréquence.

b- Donnez un invariant de boucle pour cet algorithme.

c- Prouvez sa validité et donnez sa complexité.

d- Écrire une solution récursive pour ce même problème.

e- Prouvez sa validité. Donnez sa complexité.

Exercice 6.- Soit $A[1..n]$ un tableau de n entiers positifs. On construit une liste L bidirectionnelle à l'aide des composants de A traités comme suit :

Si x est impair on l'ajoute[1] dans la liste. Si x est pair, on l'ajoute dans la liste puis on supprime de cette liste tous les éléments impairs qui le précèdent.

La liste L est initialisée avec l'entier 0.

1. Donnez les états de la liste pour le tableau A qui suit :

5	7	3	4	9	8	7	3

2. Exécuter un exemple où tous les éléments de A sont pairs et le cas où tous les éléments de A sont impairs.
3. Écrire l'algorithme qui construit L et donnez sa complexité du meilleur cas et celle du cas pire en justifiant votre réponse.

Exercice 7.- Soit D un dictionnaire (ensemble de mots) organisé en un arbre binaire de recherche. Écrire les fonctions suivantes et donnez leur complexité :

1. la recherche d'un mot dans D
2. l'insertion d'un mot dans D
3. la suppression d'un mot qui a deux fils.

Exercice 8.- La fusion de deux listes triées de m et n nœuds respectivement est en O(m+n). Qu'en est-il de la fusion de deux arbres binaires de recherche ?

1. On suppose que l'ajout en fin de liste se fait en $O(1)$.