

INFO2050 - Programmation avancée

Répétition 2: Résolutions de récurrences et sommations

Jean-Michel BEGON

02 octobre 2015

Exercice 1

Soit un tableau de N entiers où chaque entier de l'intervalle $1..N$ apparaît exactement une fois, à l'exception d'un entier apparaissant 2 fois et d'un entier manquant. Proposer un algorithme linéaire pour trouver l'entier manquant, en utilisant au plus $O(1)$ d'espace mémoire supplémentaire.

Exercice 2

Trouver une solution analytique pour

(a) $\sum_{k=1}^{\infty} (2k+1)x^{2k}$ (avec $|x| < 1$);

(b) $\sum_{i=x}^y (2i+1)$;

(c) $\sum_{i=0}^{\infty} \sum_{j=1}^n \left(\frac{j}{j+2}\right)^i$.

(d) $\sum_{k=0}^n k^2 4^k$ (Suggestion: par perturbation)

Exercice 3

Trouver une solution analytique pour la récurrence suivante:

$$T(n) = \begin{cases} 2, & \text{if } n = 0 \\ \frac{1}{n} \sum_{p=0}^{n-1} T(n-1-p)T(p), & \forall n > 1 \end{cases} \quad (1)$$

Exercice 4

Trouver une solution analytique pour la récurrence suivante:

$$\begin{cases} T(1) = 1, \\ nT(n) = (n-2)T(n-1) + 2, \quad \forall n > 1 \end{cases} \quad (2)$$

Exercice 5

Trouver une solution analytique pour la récurrence suivante:

$$T(n) = \begin{cases} 13, & \text{if } n = 1. \\ 2T(n/8) + 21n, & \forall n > 1. \end{cases} \quad (3)$$

Exercice 6

Soit la récurrence (où $n > 1$ est une puissance de 3)

$$T(n) = \begin{cases} 0, & \text{if } n = 1. \\ 6T(n/3) + 2n, & \forall n > 1. \end{cases} \quad (4)$$

- (a) Trouver une solution analytique exacte à la récurrence.
- (b) Vérifier que la solution satisfait à la borne obtenue par le *master theorem*.

Exercice 7

Pour chacun des pseudo-codes suivants, déterminer ce que fait l'algorithme, puis la complexité asymptotique.

CODE1(n)

```
1  if  $n == 0$ 
2      return 1
3  else
4      return CODE1( $n - 1$ ) + CODE1( $n - 1$ )
```

CODE1'(n)

```
1  if  $n == 0$ 
2      return 1
3  else
4      return 2 * CODE1'( $n - 1$ )
```

```

CODE2(n)
1  if n == 0
2      return ""
3  else
4      tmp = CODE2(n/2)
5      if n%2 == 0
6          return tmp + tmp
7      else
8          return tmp + tmp + "x"

```

```

CODE3(A, k)
1  for i = 1 to A.length
2      if A[i] == k
3          return i
4  return -1

```

Exercice 8

Soit une chaîne de caractères S de longueur N . Quelle serait la complexité minimale d'un algorithme qui imprime à l'écran:

- (a) Toutes les sous-chaînes contigües de longueur k ($0 \leq k \leq N$) de S ?
- (b) Toutes les sous-chaînes non nécessairement contigües de S ?

Bonus

Bonus 1

Le premier problème du projet Euler ne nécessite pas l'utilisation de l'ordinateur. Il s'énonce comme suit:

“Si on liste tous les naturels inférieurs à 10 qui sont multiples de 3 ou de 5, on obtient 3, 5, 6 et 9. La somme de ceux-ci est 23. Trouver la somme de tous les multiples de 3 ou de 5 inférieure à 1000.”