Florian Chivé
HXIV

# DS1

## *Problème 1*

Question 1

a) Les deux références **cur** et **cur_max** stockent chacune un unique entier.
La complexité spatial est donc bien en O(1).

b) Le programme se compose de deux boucles imbriquées sur les indexes de la liste. La complexité est donc :
$T(n) = \Theta(n^2)$

Question 2

a) Si l'on se donne un tableau de longueur différente de 1.
On note $s_1$, $msg_1$, $ms_1$ et $msd_1$ le résultat de l'appel récursif sur la première moitié, et $s_2$, $msg_2$, $ms_2$, $msd_2$ le résultat sur la seconde moitié.
La somme totale vaut :
$s = s1 + s2$

Pour *msg*, il n'y a que deux possibilités : soit le sous-tableau maximal est inclus dans la première moitié, soit il contient la première moitié.

Dans le premier cas on a alors $msg = msg_1$ par définition de $msg_1$.

Sinon, on a $msg = s_1 + \lambda$, avec $\lambda$ une somme d'éléments consécutifs de la deuxième moitié commençant à gauche. Par maximalité de *msg* et par définition de $msg_2$, on a là encore $\lambda = msg_2$ d'où :

$msg = \max(msg_1, s_1 + msg_2)$

De la même manière, on a :

$msd = \max(msd_2, msd_1 + s_2)$

Florian Chivé
HXIV

Quant à *ms*, on a trois possibilités, en notant **T** le sous-tableau correspondant à cette somme.

- Soit **T** est inclus dans la première moitié du tableau d'où $ms = ms_1$
- Soit **T** est inclus dans la deuxième moitié du tableau, et $ms = ms_2$
- Soit **T** contient des éléments des deux moitiés. Les éléments de la première moitié vont alors jusqu'à la fin de celle-ci, les éléments de la deuxième commençant au début de cette dernière. Par définition et maximalité, on a là encore :
  $ms = msd_1 + msg_2$ d'où :

$ms = \max(ms_1, \max(ms_2, msd_1 + msg_2)$

b) L'appel récursif à partir d'une donnée de longueur $n = j - i > 0$, dans le cas où $n$ est pair, correspond à :

- 2 appels récursifs sur une donnée de taille $\frac{n}{2}$
- Des comparaisons (appels à max) de complexité constante

On obtient donc l'équation de complexité suivante :

$$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$$

Pour tout $k \in \mathbb{N}$, on pose $u_k = T(2^k)$

On suppose $u_0 = 0$

Soit $j \geq 1$. On a :

$$T(2^j) = 2T(2^{j-1}) + O(1)$$
$$\Leftrightarrow u_j = 2u_{j-1} + O(1)$$
$$\Leftrightarrow \frac{u_j}{2^j} = \frac{u_{j-1}}{2^{j-1}} + O(\frac{1}{2^j})$$

Par récurrence, on obtient, pour $k \in \mathbb{N}$,

Florian Chivé
HXIV

$$\frac{u_k}{2^k} = \sum_{j=1}^{k} O(\frac{1}{2^j})$$

$$\Leftrightarrow u_k = 2^k \sum_{j=1}^{k} O(\frac{1}{2^j})$$

Or $\sum_{j=1}^{k} O(\frac{1}{2^j}) = O(1)$ d'où :

$$u_k = O(2^k)$$

$$T(n) = O(n)$$

---

## Problème 2

---

Question 1

b)

On effectue une boucle simple entre les indices i et j. La complexité de cette fonction est donc $\Theta(j - i + 1)$

c) On effectue une boucle sur les indices. D'après la formule de complexité des occurrences obtenue, on a, dans le pire des cas :

$$T(n) = \sum_{i=0}^{n-1} \Theta(n - i + 1)$$

On obtient donc :

$$T(n) = \Theta(n^2)$$

Question 2

a) On raisonne par contraposée : supposons que $x$ n'est majoritaire ni dans $a_1$ ni dans $a_2$.
On note $n$ la longueur de $a$. $n = j - i + 1$
Soit $o_1$ le nombre d'occurrences de $x$ dans $a_1$, $o_2$ dans $a_2$ et $o$ dans $a$.
On a : $o = o_1 + o_2$

Florian Chivé
HXIV

Par hypothèse, $x$ n'est majoritaire ni dans $a_1$ ni dans $a_2$ :

$o_1 < \frac{n}{4}$
$o_2 < \frac{n}{4}$
D'où $o < \frac{n}{2}$
Ainsi, si $x$ n'est majoritaire ni dans $a_1$ ni dans $a_2$, alors $x$ n'est pas majoritaire dans $a$.

c)
Chaque appel pour une donnée de taille $n > 1$ entraîne deux appels pour des données de taille $\frac{n}{2}$. De plus, dans le pire des cas, on fait appel à une fonction de complexité $O(n)$ deux fois.
La complexité est donc :

$$T(n) = T\left(\frac{n}{2}\right) + O(n)$$

On peut donc, d'après le cours, estimer la complexité :

$$T(n) = O(n * log_2 n)$$

Question 3

a) Soit $o_a$ le nombre d'occurrences de $a$ dans **T**. Comme $a$ est majoritaire, on a :
$o_a < \frac{n}{2}$
Il ne reste donc que $n - o_a$ cases ne contenant pas $a$ : c'est donc un majorant du nombre d'occurrence des autres éléments de **T**.
Pour $c_a = o_a$, $a$ est un postulant de **T**.

b) Si $a$ est un postulant de **T**, alors tout autre élément de T apparaît au plus $n - c_a$ fois. Comme $c_a > \frac{n}{2}$, $n - c_a < \frac{n}{2}$
Il ne peut donc y avoir d'élément distinct de $a$ qui soit majoritaire.
Si $a$ est un postulant de **T**, alors aucun autre élément de **T** n'est majoritaire.

c) Par exemple, dans le tableau [|1 ; 2 ; 3|], 3 est un postulant pour $c_3 = 2$, mais 3 n'est pas majoritaire. A l'inverse, [|1 ; 2|] n'admet aucun postulant.

d) 1) Comme $x$ n'est pas majoritaire dans $\mathbf{T}_D$, $x$ y apparaît au plus $\left\lfloor\frac{n}{4}\right\rfloor$ fois.

Florian Chivé
HXIV

*x* apparaît donc au plus $c_x$ + $\left\lfloor\frac{n}{4}\right\rfloor$ fois dans **T**. On a : $c_x$ > $\left\lfloor\frac{n}{4}\right\rfloor$, d'où $c_x$ + $\left\lfloor\frac{n}{4}\right\rfloor$ > $\frac{n}{2}$

n – ($c_x$ + $\left\lfloor\frac{n}{4}\right\rfloor$) = $\left\lceil\frac{n}{4}\right\rceil + (\frac{n}{2} - c_x)$

Tout élément de x apparaît au plus n/4 fois dans TD, et n/2 – cx fois das TG, et donc n/4 + (n/2 – cx) fois dans T.

*x* est donc un postulant de T pour $c_x$ + $\left\lfloor\frac{n}{4}\right\rfloor$.

2) x apparaît au plus cx1 fois dans TG, et cx2 fois dans TD, donc cx1 + cx2 fois dans T. cx1 > n/4 et cx2 > n/4, d'où cx1 + cx2 > n/2

Chaque élément distinct de x apparaît au plus n/2 – (cx1 + cx2) fois dans T.

*x* et donc un postulant de B pour $cx_1 + cx_2$

3)

A) *x* apparaît au plus cx fois dans TG et n/2 – cy fois dans TD, et donc n/2 + cx – cy fois dans T

Y apparaît au plus n/2 + cy – cx fois dans T.

Tout autre élément apparaît au plus n – cx – cy fois dans T.

Ainsi, aucun élément ne peut apparaître plus de n/2 fois dans T si cx = cy

Ainsi, si $c_x = c_y$, alors **T** n'admet pas d'élément majoritaire.

B)

Si cx > cy, n/2 + cx – cy > n/2

Y apparaît au plus n – (n/2 + cx – cy) fois

Tout autre élément apparaît au plus n – cx – cy fois dans T, avec n – cy < n/2 + cy car cy > n/2

*X* est donc un postulant de T pour n/2 + cx – cy.

f) Lorsqu'on fait un appel récursif, on appelle deux fois avec une donnée réduite de motié. Les autres opérations sont de complexité fixée.

On a donc la relation :

$$T(n) = 2T\left(\frac{n}{2}\right) + O(1)$$

Florian Chivé
HXIV
On reconnaît une équation de la forme :

$$T(n) = aT\left(\frac{n}{2}\right) + O(n^b)$$

Pour a = 2 et b = 0. On a donc :

T(n) = O(n). La fonction est de complexité linéaire.

h) Les fonctions *postulant* et *occurrences* étant de complexité O(n), la fonction majoritaire3 est linéaire.