Série 2

1 Complexités temporelles

Estimez la complexité temporelle des algorithmes ci-dessous (repris en partie de la dernière série) en fonction du paramètre n et en utilisant la notation $\Theta(\cdot)$.

algorithme 1

entrée : *liste L de nombres entiers, de taille n, nombre entier M*
sortie : *nombre entier positif ou nul s*

$i \longleftarrow 1$
Tant que $i \leq n$ **et** $L(i) \leq M$
  $i \longleftarrow i + 1$
$s \longleftarrow i - 1$
Sortir : s

algorithme 2

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

$s \longleftarrow 0$
Pour *i allant de 1 à $2n$*
  $s \longleftarrow s + i$
Sortir : s

algorithme 3

entrée : *liste L de nombres entiers, de taille n, nombre entier positif x*
sortie : *oui/non*

Pour *i allant de 1 à $n - 1$*
  Pour *j allant de $i + 1$ à n*
    Si $|L(j) - L(i)| > x$
      Sortir: *oui*
Sortir : *non*

algorithme 4

entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

$s \longleftarrow 0$
$j \longleftarrow 1$
Pour *i allant de 1 à n*
  Tant que *j n'est ni un multiple de 5 ni un multiple de 7*
    $j \longleftarrow j + 1$
  $s \longleftarrow s + j$
  $j \longleftarrow j + 1$
Sortir : s

2 Création d'algorithmes

a) Ecrivez un algorithme qui calcule la *moyenne arithmétique* d'une liste L de n nombres réels:

$$m_A = \frac{L(1) + L(2) + \cdots + L(n)}{n}$$

b) Ecrivez maintenant un algorithme qui *utilise l'algorithme précédent comme sous-algorithme* pour calculer la *moyenne géométrique* d'une liste L de n nombres réels positifs:

$$m_G = (L(1) \cdot L(2) \cdots L(n))^{1/n}$$

Indication: Calculer $\log_2(m_G)$ en utilisant le fait que $\log_2(a \cdot b) = \log_2(a) + \log_2(b)$ et $\log_2(a^b) = b \log_2(a)$.

c) Ecrivez un algorithme qui calcule le produit des deux plus grands nombres d'une liste L de n nombres reéls positifs (par exemple, si $L = (3, 6, 18, 12, 7)$ et $n = 5$, alors la sortie doit être $12 \times 18 = 216$).

d) Quelle est la complexité temporelle de votre dernier algorithme en fonction de la taille n de la liste L? (utiliser la notation $\Theta(\cdot)$)

3 Trier un tableau

a) Estimez la complexité temporelle de l'algorithme de tri par insertion vu au cours, en fonction de la taille n de la liste L à trier (utiliser la notation $\Theta(\cdot)$). Rappelez-vous que la complexité temporelle d'un algorithme est définie dans notre cours comme le nombre d'instructions exécutées par l'algorithme *dans le pire des cas*, i.e., pour les pires données d'entrée possibles.

b) Soit maintenant n un nombre entier positif et A un tableau de nombres entiers, de dimensions $2 \times n$. Si $n = 7$, un tel tableau pourrait être par exemple:

$$A = \begin{pmatrix} 4 & 3 & 11 & 8 & 12 & 14 & 7 \\ 5 & 2 & 12 & 13 & 15 & 5 & 3 \end{pmatrix}$$

On aimerait trier ce tableau de façon à ce que dans chaque ligne et dans chaque colonne, les nombres soient rangés dans l'ordre croissant (de gauche à droite et de haut en bas, respectivement). Une version triée du tableau ci-dessus est par exemple:

$$A' = \begin{pmatrix} 2 & 3 & 4 & 5 & 8 & 11 & 12 \\ 3 & 5 & 7 & 12 & 13 & 14 & 15 \end{pmatrix}$$

mais contrairement au cas d'une liste, qui n'a qu'une seule version triée, il y a ici plusieurs versions triées possibles du tableau A (par exemple, le 7 et le 8 peuvent être intervertis ci-dessus et le tableau reste trié).

Ecrivez un algorithme qui prenne en entrée le tableau A et sa dimension horizontale n, et dont la sortie soit une version triée du tableau A. Votre algorithme devra utiliser le sous-algorithme **tri par insertion**(L, n) d'une liste L de taille n.

Notation: $A(i, j)$ désigne l'élément de la i^e ligne et j^e colonne du tableau A (dans l'exemple ci-dessus: $A(2, 4) = 13$). On peut également utiliser la notation $A(1)$ et $A(2)$ pour désigner respectivement les première et seconde lignes du tableau A.

c) Quel est la complexité temporelle de votre algorithme? (utiliser la notation $\Theta(\cdot)$)

4 Pour le plaisir: une superstar arrive dans une réception mondaine*

Vous vous trouvez à une réception mondaine avec n autres personnes que vous ne connaissez pas et qui ne se connaissent pas non plus entre elles. Soudain, un bruit court qu'une superstar est arrivée à la réception, que tout le monde connaît sauf vous, apparemment. De son côté, la superstar ne connaît personne à la réception.

Votre tâche est d'identifier quelle est la superstar en posant au plus n questions du type "Est-ce que telle personne connaît telle autre personne?". Comment allez-vous procéder?