Aix-Marseille-Université

Test 4 UE Algorithmie et programmation 2026

Contrôle L1

11-02-2026

Les réponses sont à donner directement sur le sujet. Un espace est réservé pour chaque réponse.

Nom/Prenom	

Max et Min

On considère le problème suivant :

Nom :	MIN
Entrées :	un tableau T de réels
Sortie :	l'élément minimum du tableau

[**1**] Difficulté : ★ ☆ ☆

Démontrez que l'algorithme suivant permet de résoudre le problème **MIN** :

algorithme min(T: [réel]) → réel:
  $i^\star$:= entier ← 0
  pour chaque (i := entier) de [1 .. T.longueur[**:**
    si $T[i^\star] > T[i]$ **:**
      $i^\star \leftarrow i$
  rendre $T[i^\star]$

Un invariant de boucle est : "à la fin de l'itération i, $T[i^\star] \leq T[j]$ pour tout $j \leq i$.".

[**2**] Difficulté : ★ ☆ ☆

Donnez (et justifiez en utilisant le cours) la exité du problème algorithmique **MIN**.

En notant n la longueur du tableau passé en entrée la complexité du problème est en $\Theta(n)$:

- Il faut parcourir au moins une fois toutes les cases du tableau : la complexité est en $\Omega(n)$
- l'algorithme de la question 1 le résout : la complexité est en $\mathcal{O}(n)$

[**3**] Difficulté : ★ ☆ ☆

La complexité de l'algorithme `min(T: [réel])` → `réel` revient à compter le nombre de comparaisons effectuée. Combien en fait-il exactement ?

Il en fait $n-1$ avec n la longueur du tableau passé en entrée.

[**4**] Difficulté : ★ ★ ★

Donnez un argument (je ne demande pas de preuve complète) pourquoi tout algorithme résolvant le problème **MIN** devra faire au moins autant de comparaisons que le nombre effectué par l'algorithme `min(T: [réel]) → réel` ?

Pour que l'on puisse être sur que $T[i]$ est plus petit que $T[i^\star]$, il faut que l'algorithme ait comparé des éléments pour permettre de faire une chaîne de comparaison de $T[i^\star]$ à $T[i]$: $T[i^\star] \leq \cdots \leq T[i^{(k)}] \leq \cdots \leq T[i]$. Sinon, on pourrait prendre tous les indices que l'on ne peut pas comparer à $T[i^\star]$ et les diminuer d'une constante assez grande pour que leurs valeurs soient toutes strictement plus petite que $T[i^\star]$. L'algorithme ne pourrait pas voir la différence et rendrait toujours $T[i^\star]$ ce qui serait faux.

Comme ceci est vrai pour tout $i \neq i^\star$, il faut au moins $n-1$ comparaisons.

[**5**] Difficulté : ★ ★ ☆

Soit l'algorithme `minmax(T: [réel]) → [réel]` qui effectue les opérations suivantes :

1. ordonne les éléments $T[2i]$ et $T[2i+1]$ de telle sorte que $T[2i] \leq T[2i+1]$ à la fin de cette opération pour tout $i \geq 0$
2. cherche le min $m1$ des éléments d'indices paires de T
3. cherche le max $m2$ des éléments d'indices impaires de T
4. rend $[m1, m2]$

Montrez que l'algorithme `minmax(T: [réel]) → [réel]` rend le minimum et le maximum de T

comme $T[2i] \leq T[2i+1]$ pour tout i à la fin de la première étape :

- le minimum du tableau est forcément à un indice pair
- le maximum du tableau est forcément à un indice impair

[**6**] Difficulté : ★ ★ ☆

Combien de comparaisons effectue l'algorithme `minmax(T: [réel]) → [réel]` ?

Au total $\frac{3}{2} \cdot n$ comparaisons :

- $n/2$ pour ornonner les éléments $T[2i]$ et $T[2i+1]$ (une comparaison par i)
- $n/2 - 1$ pour trouver le min
- $n/2 - 1$ pour trouver le max
- 1 pour comparer le min avec le dernier élément si la longueur du tableau est impaire
- 1 pour comparer le max avec le dernier élément si la longueur du tableau est impaire

[**7**] Difficulté : ★ ★ ★

Conclusion ?

Il faut moins de comparaisons pour trouver le min et le max que 2 fois le nombre pour trouver uniquemnt le min.

[**8**] Difficulté : ★ ★ ★

Implémentez l'algorithme `minmax(T: [réel]) → [réel]`

algorithme min(T: [réel]) → réel:
  si $T[0] > T[1]$ **:**
    $T[0], T[1] \leftarrow T[1], T[0]$
  $i^\star$:= entier ← 0
  $j^\star$:= entier ← 1

  pour chaque (i := entier) de [1 .. T.longueur // 2[**:**
    si $T[2i] > T[2i+1]$ **:**
      $T[2i], T[2i+1] \leftarrow T[2i+1], T[2i]$
    si $T[i^\star] > T[2i]$ **:**
      $i^\star \leftarrow i$
    si $T[j^\star] < T[2i+1]$ **:**
      $j^\star \leftarrow 2i+1$
  si (T.longueur mod 2 == 1) **ET** $T[i^\star] > T[T.\text{longueur} - 1]$**:**
    $i^\star \leftarrow T$.longueur − 1
  si (T.longueur mod 2 == 1) **ET** $T[j^\star] < T[T.\text{longueur} - 1]$**:**
    $i^\star \leftarrow T$.longueur − 1

  rendre $[T[i^\star], T[j^\star]]$