TD d'Éléments d'Algorithmique n° 1

Exercice 1. *Trouver l'erreur.*

L'algorithme suivant, censé trouver le deuxième plus grand élément d'un tableau d'entiers, n'est pas correct. Montrez-le sur un exemple et corrigez l'erreur.

```
Max2(T)
entree: un tableau d'entiers T de taille n > 1
sortie: le deuxieme plus grand element de T
variables: max, max2: entier

1  si T[0]>T[1] alors max <- T[0], max2 <- T[1]
2               sinon max <- T[1], max2 <- T[0]
3  fin si
4  pour i allant de 2 a n-1 faire
5    si T[i] > max alors max <- T[i]
6                  sinon si T[i]> max2 alors max2 <- T[i]
7                        fin si
8    fin si
9  fin faire
10 renvoyer max2
```

Exercice 2. *Min et max.*

On souhaite trouver le plus petit et le plus grand élément d'un tableau d'entiers donné.

1. Écrivez l'algorithme simple qui résout le problème en effectuant, pour un tableau de n éléments, $2(n-1)$ comparaisons. Exécutez à la main votre algorithme sur le tableau

$$\mathsf{T} = \begin{array}{|c|c|c|c|c|c|} \hline 2 & 1 & \text{-3} & 5 & 4 & 8 \\ \hline \end{array}$$

2. Un algorithme plus efficace est celui qui, itérativement :
   - compare entre eux deux éléments consecutifs du tableau.
   - compare le plus petit des deux au *min* courant, et met éventuellement à jour le *min* courant.
   - compare le plus grand des deux au *max* courant, et met éventuellement à jour le *max* courant.

   Écrivez cet algorithme. Exécutez à la main votre algorithme sur le tableau T ci-dessus. Combien de comparaisons sont effectuées cette fois, en fonction de la longueur du tableau ?
3. Peut on faire mieux ? Que se passe-t-il, par exemple, si on applique la méthode du point précédent à des paquets de trois, au lieu de deux, éléments consecutifs du tableau ?

Exercice 3. *Intervalles d'entiers.*

On dispose des fonctions min et max qui renvoient respectivement le plus petit et le plus grand élément d'un tableau d'entiers donné.

1. Écrivez une fonction contientUnique qui prend en argument un tableau d'entiers T et un entier x, et qui retourne 1 si T contient x une unique fois, et qui retourne 0 sinon.

2. Écrivez une fonction verifieIntervalle qui prend en argument un tableau d'entiers T et deux entiers a et b (avec $a \leq b$) et qui retourne 1 si T contient chaque entier entre a et b (a et b compris) une unique fois, et que T ne contient aucun autre entier, et qui vaut 0 sinon. Autrement dit, cette fonction vérifie que T correspond à l'intervalle $[a, b]$. Cette fonction utilisera contientUnique.
3. Écrivez une fonction verifieIntervalle2 qui prend en argument un tableau d'entiers T et qui retourne 1 si T correspond à un intervalle (c'est-à-dire s'il existe a et b tels quel T contient chaque entier entre a et b, une unique fois, mais aucun autre) et qui retourne 0 sinon. Cette fonction utilisera verifieIntervalle, min et max.

Exercice 4. *Ordre alphabétique.*

Dans cet exercice, on considère qu'un mot (français) est un tableau de caractères, et que l'on peut connaître l'ordre (alphabétique) de deux caractères (en utilisant les opérateurs <, <=, =, >= et >). Par exemple, on a 'a' < 'b', 'b' = 'b', 'c' > 'b'. En revanche, on ne peut pas comparer directement deux mots.

1. Écrivez une fonction inf qui prend en argument deux mots et qui retourne -1 si le premier mot est avant le second dans l'ordre alphabétique (ou lexicographique, c'est-à-dire l'ordre du dictionnaire), 0 si les deux mots sont les mêmes, et +1 sinon. Exécutez à la main cette fonction suir les tableaux de caractères | t | a | b | l | e | et | t | a | b | l | e | a | u |.
2. Écrivez une fonction min (utilisant inf) qui prend en argument un tableau de mots et qui retourne le plus petit mot pour l'ordre alphabétique.

Exercice 5. *Médiane.*

La médiane d'un tableau T à n éléments distincts avec n impair, est l'élément x de T tel que exactement $(n-1)/2$ éléments de T sont < x.

1. Écrivez une fonction compteInf qui prend en argument un tableau d'entiers T et un entier x, et qui retourne le nombre d'éléments de T qui sont < x.
2. Écrivez un algorithme (utilisant cette fonction) qui trouve la médiane d'un tableau T à n éléments distincts avec n impair.
3. Quelle est la complexité de cet algorithme en termes du nombre de comparaisons effectuées, dans le pire des cas, en fonction de n ?
4. Donnez un exemple de tableau où la borne trouvée au point précédent est atteinte, et dérouler l'algorithme du point 2 de cet exercice sur le tableau donné.