

EA3

Partiel du jeudi 3 novembre 2016

Durée : 1 heures 30

Aucun document autorisé

Appareils électroniques éteints et rangés

Préliminaires : *Ce sujet est constitué de 4 exercices qui peuvent être traités dans l'ordre de votre choix. Le barème est donné à titre indicatif. Il est conseillé de lire l'intégralité du sujet avant de commencer. Il est bien entendu préférable de ne faire qu'une partie du sujet correctement plutôt que de tout bâcler.*

Exercice 1 : un peu de lecture (3 points)

```

1 quoi(T : tableau de n entiers, i : entier):
2   if(i < n/2){
3     tmp <- T[i]
4     T[i] <- T[n-1-i]
5     T[n-1-i] <- tmp
6     quoi(T, i+1)
7   }

```

1. Exécuter l'algorithme `quoi` sur le tableau $T=\{3, 6, 5, 2, 8, 1, 4\}$ et $i=0$ en déroulant la pile d'exécution.
2. Écrire un algorithme itératif qui produit le même tableau en sortie que `quoi`.

Exercice 2 : algorithme coton (7 points)

On donne l'algorithme suivant `ouate` avec `trier(T)` une fonction qui trie dans l'ordre croissant un tableau d'entiers T .

```

1 ouate(T : tableau de n entiers, x : entier):
2   trier(T)
3   i <- 0   j <- n-1
4   while(i < j){
5     if(T[i]+T[j] = x) return T[i],T[j]
6     else{
7       if(T[i]+T[j] < x) i <-i+1
8       else if(T[i]+T[j] > x) j <- j-1
9     }
10  }
11  return -, -

```

1. Exécuter l'algorithme `ouate` sur le tableau $T=\{3, 6, 5, 2, 8, 1, 4\}$ et $x=8$.
2. Donner un pseudo-code de l'algorithme de tri fusion pour la fonction `trier(T)`. Rappeler sa complexité et ses propriétés (est-il stable, en place?).
3. On suppose le tableau T trié par ordre croissant. Soit la propriété suivante : *S'il existe deux valeurs distinctes de T , u et v telles que $u+v=x$, alors $T[i] \leq u, v \leq T[j]$.* On veut montrer que cette propriété est un invariant de la boucle `while` de `ouate`.
 - a. Montrer que la propriété est vraie à l'entrée dans la boucle `while` de `ouate`, c'est-à-dire lorsque $i=0$ et $j=n-1$.
 - b. Montrer que si on suppose que la propriété est vraie pour un certain i et un certain j au début d'un passage dans la boucle, alors elle est toujours vraie à la fin du passage dans la boucle pour les nouvelles valeurs de i et j .
4. Expliquer ce que fait l'algorithme en justifiant.

Exercice 3 : quelles sont tes racines ? (4 points)

Soit une fonction f continue, strictement croissante sur un intervalle $[a, b]$, $a \leq b$, tel que $f(a) < 0$ et $f(b) > 0$. Alors f possède une unique racine r sur $[a, b]$ ($f(r) = 0$). On souhaite trouver un encadrement $[i, j]$, $i \leq j$, pour r tel que $[i, j]$ soit inclus dans $[a, b]$, $j - i \leq 1$, $f(i) \leq 0$ et $f(j) \geq 0$. Par exemple, pour la fonction $f(x) = x^2 + 3x - 7$, $a = -1$, $b = 3$, l'algorithme donne $i = 1$ et $j = 2$.

1. Soit $m = \frac{a+b}{2}$. À quelle condition sur $f(m)$, peut-on affirmer que $r \in [a, m]$? Et que $r \in [m, b]$?
2. En vous inspirant de l'algorithme de dichotomie, écrire un algorithme qui détermine l'intervalle $[i, j]$ étant donné une fonction f continue, strictement croissante sur un intervalle $[a, b]$ tel que $f(a) < 0$ et $f(b) > 0$.

Exercice 4 : tri coquillage (6 points)

1. Soit un tableau T de n entiers et un entier p , $1 \leq p < n$. écrire l'algorithme `triPartiel(T, p)` qui est une réécriture de l'algorithme de tri par insertion en ne triant dans T que les éléments d'indices $0, p, 2p, \dots$.
Par exemple si $T = \{6, 2, 9, 12, 5, 3, 7, 1, 11, 4, 8, 10\}$ et $p = 3$, alors `triPartiel(T, p)` doit trier les éléments d'indices $0, 3, 6$ et 9 et modifie donc le tableau T de la façon suivante :
 $\{4, 2, 9, 6, 5, 3, 7, 1, 11, 12, 8, 10\}$.
2. Pour trier un tableau T de n entiers, on propose d'appeler l'algorithme `triPartiel(T, p)` pour la suite de valeurs de p suivante : $\lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{8} \rfloor, \dots, 1$.
Dans l'exemple de la question 1, la suite de valeurs de p est donc $6, 3, 1$.
 - a. Écrire l'algorithme `tri` qui suit le principe ci-dessus et utilise `triPartiel`.
 - b. Pourquoi est-on sûr que cet algorithme trie bien tout tableau d'entiers passé en paramètre ?
 - c. Combien de comparaisons sont faites dans l'algorithme `tri` pour trier le tableau T donné en exemple ? Et combien pour le tri par insertion sur cet exemple ?
 - d. Pourquoi, à votre avis, ce tri peut améliorer la complexité du tri par insertion sur certaines entrées ?