Entraînement : algorithmes de tris

Pour tous les exercices, la grille d'évaluation est la suivante.

Identification de l'algorithme et complexité.

A (20)	Algorithme bien identifié / décrit et bonne complexité
C (11)	Complexité correcte mais algorithme mal identifié / décrit
D (8)	Algorithme bien identifié / décrit mais erreur dans la complexité
E (1)	Ni complexité, ni identification juste

Réalisation de la partie d'algorithme manquante.

A (20)	L'algorithme répond correctement au problème posé, il est écrit de façon claire et compacte et est de complexité optimale.
B (16)	L'algorithme contient quelques erreurs mais reste globalement juste et la complexité optimale est respectée.
C (11)	L'algorithme fonctionne globalement mais complexité non optimale.
D (8)	L'algorithme ne fonctionne pas.
E (1)	Algorithme quasi inexistant ou ne répondant pas du tout au problème posé.

Exercice 1 (Tri rapide – 5 pts)**.**

(1) Rappeler en quelques phrases le principe du tri rapide (quicksort).

En voici une implantation partielle :

```
TriRapide
Input :
    − T, un tableau de nombres
    − a, le premier indice de la zone à trier
    − b, le dernier indice de la zone à trier
Procédé :
    Si b <= a :
        Retourner
    m <− Pivot(T,a,b)
    TriRapide(T,a,m−1)
    TriRapide(T,m+1,b)
```

Pour un tableau `T` de taille 5, on appellerait la fonction de cette façon `TriRapide(T,0,4)` car 4 est le dernier indice du tableau.

La fonction `Pivot` modifie le tableau de telle sorte qu'après son passage, toutes les valeurs avant l'indice `m` doivent être plus petite ou égale à `T[m]` et toutes les valeurs après l'indice `m` doivent être plus grande que `T[m]`. Par exemple, elle pourrait modifier le tableau suivant de cette façon :

Avant le passage de Pivot :

5 2 6 3 1

Après le passage de Pivot :

3 2 1 **5** 6

L'indice `m` retourné par la fonction dans ce cas est **3**.

(2) Donner une implantation de la fonction `Pivot`

Remarque : votre fonction n'est pas obligée d'agir exactement comme dans l'exemple, elle doit seulement respecter les propriétés énoncées ci-dessus.

(3) Donner les étapes de votre propre algorithme de pivot sur le tableau donné en exemple :

5 2 6 3 1

(4) Quelle est la complexité de l'algorithme `TriRapide` dans les trois cas suivant : tableau déjà trié, meilleur des cas, en moyenne.

Solution

(1) Principe : A chaque étape on choisit un pivot v (par exemple, le premier élément du tableau), puis on place les éléments du tableau tel que tous les éléments *plus petits* que v soient sur la gauche du tableau, tous les éléments *plus grands* sur la droite et v entre la partie gauche et la partie droite. Puis on trie récursivement les parties gauches et droites.

(2) cf cours

(3) cf cours

(4) tableau déjà trié : $O(n^2)$, meilleur des cas : $O(n\log(n))$, en moyenne $O(n\log(n))$.

Exercice 2 (Tri Fusion – 5 pts).

(1) Rappeler en quelques phrases le principe du tri fusion.

En voici une implantation partielle :

```
TriFusion
Input :
    - T, un tableau de taille n
Output :
    - Un tableau trié contenant les mêmes valeurs que T
Procédé :
    Si n <= 1 :
        Retourner T
    m <- n/2
    T1 <- TriFusion(T[ :m+1])
    T2 <- TriFusion(T[m+1 :])
    Retourner Fusion(T1,T2)


Fusion
Input :
    - T1, un tableau de taille m1 supposé trié
    - T2, un tableau de taille m2 supposé trié
Output :
    - un tableau trié contenant les valeurs de T1 et T2
Procédé
    ....
```

L'écriture `T[ :m+1]` signifie une copie du tableau `T` entre les indices 0 et m inclus, `T[m+1 :]` signifie une copie du tableau `T` entre les indices $m+1$ et $n-1$ inclus.

Voici un exemple du résultat de la fonction `Fusion` :

`T1` 2 2 4 5 5

`T2` 3 4 6 7

la fonction renvoie

2 2 3 4 4 5 5 6 7

(2) Compléter le procédé de la fonction `Fusion`. Votre algorithme doit avoir une complexité $O(n)$ où $n = m1 + m2$.

(3) Donner les étapes de votre propre algorithme de fusion sur l'exemple ci-dessus.

(4) Quelle est la complexité de l'algorithme complet `TriFusion` dans le pire des cas ?

Solution

(1) L'algorithme découpe le tableau en deux parts égales et trie les deux sous-tableaux puis les fusionne en un tableau unique.

(2) cf cours

(3) cf cours

(4) Pire des cas : $O(n \log(n))$