

Programmation 2

L1 2011-2012

Travaux Pratiques 5 - Tris Efficaces

Dans ce tp, on implémente trois algorithmes de tri d'entiers.

L'algorithme Quicksort (ou tri rapide) est un algorithme de tri récursif. Nous allons le décrire pour un tri en ordre croissant :

- Si le tableau est de taille 1 ou 0, alors il est trié.
- Sinon on choisit un pivot. En général ce pivot est la valeur de la première case du tableau. Ensuite on place dans les premières cases du tableau toutes les valeurs inférieures ou égales à ce pivot. Appelons n ce nombre de cases. Dans les cases suivantes, on place toutes les valeurs supérieures à ce pivot. On tri avec le même algorithme le tableau formé par n premières cases à gauche (les valeurs inférieures au pivot). On tri avec le même algorithme le tableau formé par les cases restantes à droite.

Le tableau est trié.

Exercice 1. Tri rapide

1. Écrire une fonction `void triQsortcrois(int T[],int debut,int fin)` qui tri les éléments d'un tableau dans l'ordre croissant selon l'algorithme Quicksort.
2. Quel est la complexité de cet algorithme dans le pire des cas. Exhiber un exemple de pire cas.
3. Modifier la fonction `triQsort` en une fonction `triQsortcroiscpt` qui permet de compter le nombre de comparaison du tri.
4. Écrire une fonction `void triQsortdecroi(int T[],int debut,int fin)` qui tri les éléments d'un tableau dans l'ordre décroissant selon l'algorithme Quicksort.

On veut maintenant implémenter l'algorithme Trifusion. L'algorithme Trifusion est un algorithme de tri récursif. Nous allons le décrire pour un tri dans l'ordre croissant :

- Si le tableau est de taille 1 ou 0 alors il est trié.
- Sinon on coupe le tableau, par le milieu, en deux parties égales (à une case près). On utilise l'algorithme pour trier le tableau de gauche, et on réutilise l'algorithme pour le tableau de droite. Ensuite on fabrique un

nouveau tableau N de même taille que l'original. Les tableaux de droite et de gauche sont triés donc le plus petit élément m de l'union des deux tableaux est soit au début du tableau de gauche soit au début du tableau de droite. On déplace ce plus petit élément dans la première case du tableau N (on retire cet élément du tableau où il était). On remarque que les tableaux gauche et droite sont toujours triés. On met maintenant le plus petit élément des nouveaux tableaux droite et gauche dans la seconde case de N . On continue comme ça jusqu'à avoir épuisé tout les éléments des tableaux gauche et droite.

Le tableau N est trié on le recopie dans le tableau original.

Il existe une autre version de cet algorithme de tri sur listes chaînées qui ne nécessite pas de recopier le tableau en fin de tri.

Exercice 2. Tri fusion

Écrire une fonction `void TriFusion(int T[],int debut,int fin)` de tri d'un tableau selon l'algorithme Trifusion.

Exercice 3. tri par tas

Écrire un algorithme de tri par tas.