TP4 : Algorithmes de tri

Préambule

L'objectif de cette séance est de se familiariser avec différents algorithmes de tri vus en cours. À la fin de ce TP, vous devriez être capable de mettre en œuvre des algorithmes de tri :

- avancés (tri par tas, tri rapide, tri Shell),
- en suivant une description d'une idée.

Quelques informations importantes :

- Le langage de programmation à utiliser est le C.
- Ce TP utilise les signaux (pour interrompre les tris lorsqu'ils dépassent une certaine limite de temps), il doit être fait sous Linux.
- Des outils pour tester les fonctions sont fournis dans le fichier zip `tp03_code.zip`.
- Pour chaque exercice, exécutez les tests dans les fichiers fournis pour vérifier le code écrit.
- Il est fortement recommandé de **compiler dans le terminal** (en utilisant make).

Programme de tests et exercices suggérés par Jean-Claude Georges et adaptés pour le cours.

Nous continuerons à utiliser les fichiers du TP3 (`tp03_code.zip`). Si vous ne l'avez pas encore fait, résolvez les exercices 1 et 2 du TP3 afin de vous familiariser avec les fichiers.

1 Les tris avancés

Exercice 1 : Le tri shell (*). Déjà écrit. Testez-le. Comparez ses performances avec celles des tris précédents.

Exercice 2 : Le tri par tas (**). Le fichier `tri.c` contient déjà toutes les fonctions nécessaires pour mettre en œuvre l'algorithme de tri par tas. Complétez la fonction `tri_tas`.

Exercice 3 : Le tri rapide (**). Écrivez une fonction `partition(double t[], int n)` qui effectue la partition (les petits devant, les grands derrière) autour du pivot `t[0]` et retourne la place définitive de `t[0]`.

Écrivez une fonction `tri_rapide_1`. Vérifiez qu'elle est bien plus performante pour un tableau aléatoire, et catastrophique pour un tableau trié.

Exercice 4 : Le tri rapide sur PLaTon (P). Modifiez la fonction que vous avez écrit pour résoudre l'exercice PLaTon. Votre programme doit afficher, pour chaque appel récursif sur deux éléments ou plus, le sous-tableau correspondant après la partition, ainsi que le tableau trié à la fin du tri.

Activité PLaTon : TP4

Exercice 5 : Le tri shell sur PLaTon (P). Modifiez le tri Shell que vous avez écrit pour résoudre l'exercice PLaTon en utilisant la séquence de Hibbard. Votre programme doit afficher le tableau après chaque valeur de h.

Activité PLaTon : TP4

Exercice 6 : Combinaison du tri rapide et du tri Shell (*). Écrivez une autre fonction `tri_rapide_2` que exécute le tri rapide et utilise le tri Shell lorsque le nombre d'éléments à trier est inférieur ou égal à 6.

2 Pour aller plus loin

Exercice 7 : Le tri rapide amélioré (***). Essayez d'autres stratégies de sélection du pivot. Arrivez vous à améliorer l'algorithme dans les tableaux triés (en ordre croissante et décroissante) ?

Exercice 8 : Le tri patience (***). Mettre en œuvre le tri patience.

Le tri par patience est basé sur un jeu de cartes et trie un tableau en deux phases. Dans la première phase, nous déplaçons les éléments du tableau sur des piles (tableaux) :

- Au départ, il n'y a pas de piles. Le premier élément forme une nouvelle pile.
- Chaque élément suivant est placé sur la pile existante la plus à gauche dont le dernier élément a une valeur supérieure ou égale à la valeur du nouvel élément, ou à droite de toutes les piles existantes, formant ainsi une nouvelle pile.
- Lorsqu'il ne reste plus d'éléments à placer, la première phase se termine.

Pour trier le tableau, prenez le plus petit élément à la dernière position de l'une des piles, retirez-le de la pile et placez-le en première position du tableau. Répétez ensuite le même processus pour remplir les positions suivantes.

Modifiez également le fichier `testtri.c` pour ajouter cet algorithme au menu.

Exercice 9 : Advent of Code (en anglais). Advent of Code est un calendrier de l'Avent proposant de petites énigmes de programmation pour différents niveaux de compétence, qui peuvent être résolues dans le langage de programmation de votre choix.

Si vous avez terminé les exercices précédents et vous souhaitez vous entraîner à des exercices de programmation et de résolution de problèmes plus difficiles mais tout aussi amusants, rendez-vous à l'adresse ci-dessous.

Advent of Code