

Méthodes de Conception d'Algorithmes

C. TRABELSI & N. IZRI & M. FRANÇOIS

TP4 -- Tri rapide (Quicksort)

Tri rapide (*quicksort*)

PARTIE A (Partitionnement de LOMUTO)

- ▶ 1. Écrire une fonction `AFFICHAGE_TAB`, permettant d'afficher les éléments d'un tableau unidimensionnel donné en paramètre.
- ▶ 2. Écrire la fonction `PARTITION_LOMUTO` vue en TD. Tester votre fonction sur le tableau :
`TAB = [12, 23, 6, 5, 13, 9, 8, 5, 20, 3, 7, 11]`
Normalement si la fonction est bien écrite, après exécution le tableau `TAB` doit être :
`TAB = [6, 5, 9, 8, 5, 3, 7, 11, 20, 12, 23, 13]`
- ▶ 3. Utiliser cette fois-ci un tableau dont les valeurs sont toutes identiques, et trouver la valeur de retour de la fonction `PARTITION_LOMUTO`.
- ▶ 4. Écrire la fonction récursive `TRI_RAPIDE`. Tester votre programme sur les tableaux précédents et également d'autres tableaux.
- ▶ 5. La fonction `PARTITION_LOMUTO` effectue un parcours gauche-droite sur le tableau d'entrée. Écrire la fonction `PARTITION_LOMUTO_D_G` qui adopte une stratégie droite-gauche pendant le parcours du tableau. Utiliser cette dernière avec la fonction `TRI_RAPIDE`.

PARTIE B (Partitionnement de HOARE)

- ▶ 1. Écrire la fonction `PARTITION_HOARE` vue en TD. Tester votre fonction sur le tableau :
`TAB = [12, 23, 6, 5, 13, 9, 8, 5, 20, 3, 7, 11]`
Normalement si la fonction est bien écrite, après exécution le tableau `TAB` doit être :
`TAB = [11, 7, 6, 5, 3, 9, 8, 5, 20, 13, 23, 12]`
- ▶ 2. Pour un tableau dont les valeurs sont toutes identiques, quelle est la valeur de retour de la fonction `PARTITION_HOARE` ?
- ▶ 3. Écrire la fonction récursive `TRI_RAPIDE_2` utilisant la fonction `PARTITION_HOARE`. Tester votre programme sur les tableaux précédents et également d'autres tableaux.

PARTIE C (LOMUTO *vs* HOARE)

- ▶ 1. On considère le tableau : $TAB = [12, 23, 6, 5, 13, 9, 8, 5, 20, 3, 7, 11]$.
Calculer puis comparer le nombre de cycles d'exécution de la fonction `TRI_RAPIDE` et celle de `TRI_RAPIDE_2` avec l'outil callgrind (Réf. TD2_O).
Conclure.
- ▶ 2. On considère le tableau contenant des valeurs identiques :
 $TAB = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$.
Calculer puis comparer le nombre de cycles d'exécution de la fonction `TRI_RAPIDE` et celle de `TRI_RAPIDE_2` avec l'outil callgrind.
Conclure.
- ▶ 3. On considère cette fois-ci un tableau déjà trié :
 $TAB = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]$.
Calculer puis comparer le nombre de cycles d'exécution de la fonction `TRI_RAPIDE` et celle de `TRI_RAPIDE_2` avec l'outil callgrind.
Conclure.

PARTIE D (PROJET)

Ceux qui ont tout fini, ils peuvent travailler sur le projet.