Informatique

2
Algorithmes à boucles imbriquées

TD 2-3
Tri bulles

Exercice 1: Tri bulle

Soit une liste L de n entiers.
Soit le premier algorithme suivant :

- On parcourt la liste en inversant $L[i]$ et $L[i+1]$ chaque fois que $L[i] > L[i+1]$
- On recommence tant qu'il y a eu des échanges au parcours précédent

Question 1: Créer une fonction Parcours_1(L) qui parcours une fois L, fait les échanges si besoin et renvoie une variable valant 1 si au moins un échange a été réalisé, 0 sinon
Question 2: Créer la fonction Tri_bulle_1(L) qui réalise ce tri
Question 3: Sur un ordinateur, en faisant afficher la liste après chaque parcours, vérifier la propriété suivante : « A la fin du k-ème parcours, les k plus grands éléments sont à leur place finale » (se démontre par récurrence)

Soit le second algorithme suivant :

- On parcourt la liste $n-1$ fois en inversant $L[i]$ et $L[i+1]$ chaque fois que $L[i] > L[i+1]$
- La k-ème fois, on ne « touche pas » aux $k-1$ derniers éléments (déjà triés)

Question 4: Créer une fonction Parcours_2(L,k) (k>0) sur la base de Parcours_1 qui parcours la liste L sans toucher à ses k-1 derniers éléments en procédant aux inversions vues précédemment
Remarque : Pour vous aider un peu,

- Quand $k = 1$, la fonction ne « touche pas » au $k - 1 = 0$ derniers éléments, elle peut donc « toucher » au dernier élément, soit échanger l'avant dernier terme $L[n-2]$ et le dernier $L[n-1]$
- Quand $k = 2$, la fonction ne « touche pas » au $k - 1 = 1$ derniers éléments, elle ne « touchera » donc pas au dernier élément $L[n-1]$ lors d'un éventuel échange entre $L[i]$ et $L[i+1]$
- Etc.

Question 5: Créer la fonction Tri_bulle_2(L) qui réalise ce tri
Question 6: Sur un ordinateur, comparer l'efficacité de ces deux algorithmes

Remarquez que l'inégalité stricte $L[i] > L[i+1]$ fait que cet algorithme est stable.

Question 7: Est-il facile de se retrouver dans le pire des cas de temps d'exécution de ces algorithmes ?