

1 Trier des entiers

On considère durant cette séance de TD que l'on désire trier un tableau d'entiers par ordre décroissant. Il existe une grande quantité de méthodes de tri, le tri par insertion, par sélection, par base, par tas, par comptage, par bulle, le tri rapide, le tri fusion... Nous donnons ci-dessous quelques exemples de méthodes. Il vous est demandé d'en choisir une qui paraisse pertinente dans le cadre d'une parallélisation et d'en proposer un algorithme séquentiel.

1.1 Je trie, tu tries, il trie...

1.2 Tri bulle

Invariant : à l'itération i , les i premiers éléments du tableau sont triés par ordre décroissant.

Principe : à chaque itération, on cherche l'élément le plus grand dans le sous-tableau contenant les éléments non triés et on le place dans la case du premier élément du sous-tableau non trié. Le sous-tableau trié est plus long d'une case et le sous-tableau non trié est plus court d'une case.

Exemple :

itération 1	12	34	7	-3	23
itération 1	12	34	7	23	-3
itération 1	12	34	23	7	-3
itération 1	34	12	23	7	-3
itération 2	34	12	23	7	-3
itération 2	34	12	23	7	-3
itération 2	34	23	12	7	-3
itération 3	34	23	12	7	-3
itération 3	34	23	12	7	-3
itération 4	34	23	12	7	-3
itération 5	34	23	12	7	-3
itération 6	34	23	12	7	-3

1.3 Tri par comptage-insertion

Invariant : à l'itération i , i^e éléments du tableau occupent leur position dans le tableau trié.

Principe : en partant du premier élément, on comptabilise le nombre d'éléments qui sont plus grands que lui dans le tableau. Si ce nombre est k , on place l'entier dans la $k + 1$ e case du tableau.

Exemple :

tableau	12	34	7	-3	23
nombre d'éléments supérieurs	2	0	3	4	1
tableau trié	34	23	12	7	-3
itération 1	7	34	12	-3	23
itération 2	-3	34	12	7	23
itération 3	23	34	12	7	-3
itération 4	34	23	12	7	-3
itération 5	34	23	12	7	-3

Problème lorsque certains éléments sont égaux, il faut effectuer un test d'égalité et faire glisser l'élément jusqu'à la première case qui contient un élément plus petit que lui-même.

tableau	12	34	7	12	23
nombre d'éléments strictement supérieurs	2	0	4	2	1
tableau trié	34	23	12	12	7

1.4 Tri par insertion

Invariant : à l'itération i , le i^{e} élément est à sa place dans le tableau.

Principe : en partant du premier élément, on cherche dans le tableau le premier élément qui est plus petit que lui (tri par ordre décroissant) et on le place à cet endroit, en décalant le reste du tableau. C'est le principe de tri que chacun de nous utilise lorsqu'il range une feuille au sein d'une pile de feuilles. La position de la feuille est déterminée puis la feuille est insérée.

Exemple :

tableau de départ	12	34	7	-3	23
itération 1	34	12	7	-3	23
itération 2	34	12	7	-3	23
itération 3	34	12	7	23	-3
itération 4	34	23	12	7	-3
itération 5	34	23	12	7	-3

1.5 Fusion de deux tableaux triés.

Principe : Le premier élément du tableau fusionné sera le premier élément de l'un des deux tableaux d'entrée.

Exemple : Fusionner [1;2;5] et [3;4] :

On compare donc 1 et 3 \rightarrow 1 est plus petit : [2;5] - [3;4] \rightarrow [1]

On compare 2 et 3 \rightarrow 2 est plus petit : [5] - [3;4] \rightarrow [1;2]

On compare 5 et 3 \rightarrow 3 est plus petit : [5] - [4] \rightarrow [1;2;3]

On compare 5 et 4 \rightarrow 4 est plus petit : [5] \rightarrow [1;2;3;4]

[1;2;3;4;5]

1.6 Exercices

- 1 Ecrire l'algorithme de tri par comptage.
- 2 Soient E_1, E_2 deux tableaux triés, écrire un algorithme permettant de fusionner ces deux tableaux pour obtenir un nouveau tableau trié.
- 3 Soit E un tableau d'entiers, écrire un programme en c permettant d'intégrer les deux les deux algorithmes précédent dans un programme permettant le tri du tableau E .

Nota bene : - Ces deux algorithmes devra être implémenté en séquentiel durant la prochaine séance de TP.
- Une implémentation parallèle sera étudiée durant les séances de TD et TP qui suivra...