Les algorithmes de tri

Site Internet : **www.gecif.net**	Type de document : **Cours**	Intercalaire :	Date :

En informatique **un algorithme de tri** est un algorithme qui permet d'organiser une collection d'objets selon une relation d'ordre déterminée. On retrouve par exemple ces algorithmes dans les bases de données ou dans les moteurs de recherche. On dit qu'un algorithme de tri est **stable** s'il ne modifie pas l'ordre initial des clés identiques.

I – Le tri par sélection

Le tri par sélection est un algorithme de tri par comparaison. Cet algorithme est simple, mais considéré comme inefficace car il s'exécute en temps quadratique (il devient très vite lent pour un grand nombre de valeurs à trier). De plus le tri par sélection est **instable** car en cas de valeurs identiques dans le tableau initial leur ordre relatif peut être modifié. Sur un tableau de n éléments (numérotés de 0 à n-1), le principe du tri par sélection est le suivant :

- rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 0
- rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1
- continuer de cette façon jusqu'à ce que le tableau soit entièrement trié

Illustration avec le tableau non trié à 6 éléments [5,3,1,4,6,2] (en rouge les éléments à leur place définitive) :

Indice des éléments ➔	0	1	2	3	4	5
Recherche du plus petit élément entre 0 et 5	**5**	**3**	**1**	**4**	**6**	**2**
Permutation avec l'élément d'indice 0						
Recherche du plus petit élément entre 1 et 5						
Permutation avec l'élément d'indice 1						
Recherche du plus petit élément entre 2 et 5						
Permutation avec l'élément d'indice 2						
Recherche du plus petit élément entre 3 et 5						
Permutation avec l'élément d'indice 3						
Recherche du plus petit élément entre 4 et 5						
Permutation avec l'élément d'indice 4 : **FIN**						

Dans le tri par sélection on ne fait que des permutations utiles : après chaque permutation un nouvel élément est à sa place définitive (les éléments permutés ne sont pas forcément voisins). En revanche il faut rechercher la valeur minimale dans un tableau non trié, ce qui est très couteux en temps d'exécution (parcourt de toute la liste restante).

II – Le tri par insertion

Le tri par insertion est un algorithme de tri classique. La plupart des personnes l'utilisent naturellement pour trier des cartes à jouer. En général, le tri par insertion est beaucoup plus lent que d'autres algorithmes de tri pour traiter de grandes séquences, car sa complexité est quadratique. Le tri par insertion est cependant considéré comme l'algorithme le plus efficace sur des tableaux de petite taille. Il est aussi efficace lorsque les données sont déjà presque triées. De plus le tri par insertion est **stable** car en cas de valeurs identiques dans le tableau initial leur ordre relatif n'est pas modifié. Le principe du tri par insertion est le suivant : on fait comme si les éléments à trier étaient donnés un par un, le premier élément constituant, à lui tout seul, une liste triée de longueur 1. On range ensuite le deuxième élément pour constituer une liste triée de longueur 2, puis on range le troisième élément pour avoir une liste triée de longueur 3 et ainsi de suite jusqu'au dernier élément.

Le principe du tri par insertion est donc d'insérer à la $n^{ième}$ itération le $n^{ième}$ élément à la bonne place en le faisant redescendre jusqu'à atteindre un élément qui lui est inférieur.

Illustration avec le tableau non trié à 6 éléments [5,3,1,4,6,2] (en vert les éléments en attente) :

Indice des éléments ➔	0	1	2	3	4	5
Tableau initial ➔	**5**	**3**	**1**	**4**	**6**	**2**
Insertion de l'élément d'indice 1 à sa place						
Insertion de l'élément d'indice 2 à sa place						
Insertion de l'élément d'indice 3 à sa place						
Insertion de l'élément d'indice 4 à sa place						
Insertion de l'élément d'indice 5 à sa place : **FIN**						

Dans le tri par insertion on fait beaucoup de petites permutations d'éléments voisins (décalage de tous les éléments de gauche pour insérer la nouvelle clé à sa place) mais en revanche il n'y a pas de recherche à faire dans le tableau.

III – Le tri à bulles

Le tri à bulles est un algorithme de tri classique. Son principe est simple, et il est très facile à implémenter. L'algorithme parcourt le tableau, et dès que deux éléments consécutifs ne sont pas ordonnés, les permute. Après un premier passage, on voit que le plus grand élément se situe bien en fin de tableau. On peut donc recommencer un tel passage, en s'arrêtant à l'avant-dernier élément, et ainsi de suite. Au $n^{ème}$ passage on fait remonter le $n^{ème}$ plus grand élément du tableau à sa position définitive, un peu à la manière de bulles qu'on ferait remonter à la surface d'un liquide, d'où le nom d'algorithme de tri à bulles.

Illustration avec le tableau non trié à 6 éléments [5,3,1,4,6,2] (en rouge les éléments à leur place définitive) :

Indice des éléments ➔	0	1	2	3	4	5
Tableau initial ➔	**5**	**3**	**1**	**4**	**6**	**2**
Comparaison et permutation des indices 0 et 1						
Comparaison et permutation des indices 1 et 2						
Comparaison et permutation des indices 2 et 3						
Comparaison et permutation des indices 3 et 4						
Comparaison et permutation des indices 4 et 5						
Comparaison et permutation des indices 0 et 1						
Comparaison et permutation des indices 1 et 2						
Comparaison et permutation des indices 2 et 3						
Comparaison et permutation des indices 3 et 4						
Comparaison et permutation des indices 0 et 1						
Comparaison et permutation des indices 1 et 2, etc.						

Le principe du tri à bulles est donc de comparer deux à deux les éléments e1 et e2 consécutifs d'un tableau et d'effecteur une permutation si e1 > e2. On continue de trier jusqu'à ce qu'il n'y ait plus de permutation : dans le tri à bulles on fait beaucoup de petites permutations d'éléments voisins. De plus le tri à bulles est **stable** car en cas de valeurs identiques dans le tableau initial leur ordre relatif n'est pas modifié.

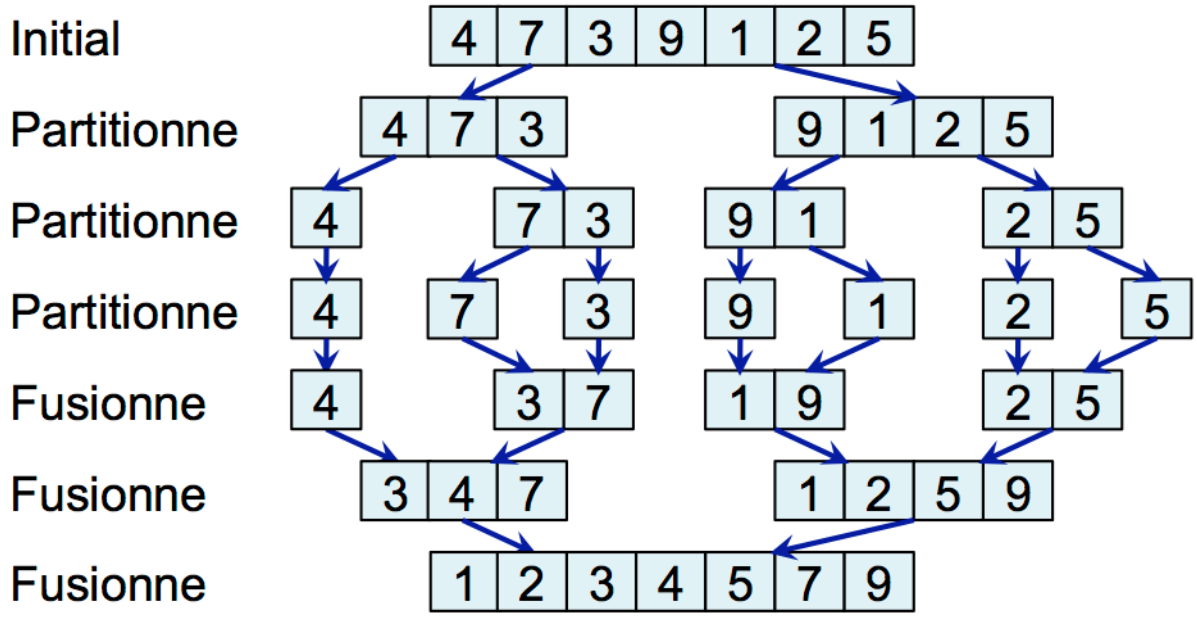
IV – Le tri fusion

L'algorithme du tri fusion repose sur la méthode « diviser pour mieux régner ». Le principe du tri fusion est le suivant :

- la liste est divisée en 2 sous-listes
- chaque sous-liste est à nouveau diviser en 2, jusqu'à obtenir seulement des listes à 1 élément
- ensuite on fusionne 2 éléments pour former une **liste triée à 2 éléments**
- il suffit ensuite de fusionner deux à deux **les listes triées** pour obtenir la liste complète triée

L'efficacité du tri fusion vient du fait que pendant la phase de fusion les listes sont déjà triées : il suffit alors de comparer seulement leurs premiers éléments pour savoir quel est le prochain élément de la liste fusionnée.

Illustration avec le tableau non trié à 7 éléments [4,7,3,9,1,2,5] :



Avec le tri fusion, à aucun moment on a parcourt toute une liste complète (contrairement au tri par insertion), et à aucun moment on doit recherche une valeur minimale dans une liste non triée (contrairement au tri par sélection). De plus le tri fusion est **stable** car en cas de valeurs identiques dans le tableau initial leur ordre relatif n'est pas modifié.