

Algorithmique et structures de données

TD1 – Tris sur tableaux

Lamia BENAMARA

Consigne :

Pour les séances TD, tous les algorithmes demandés sont à écrire en langage algo, consulter la fiche algo disponible sur le drive pour voir la syntaxe ainsi que des exemples complets.

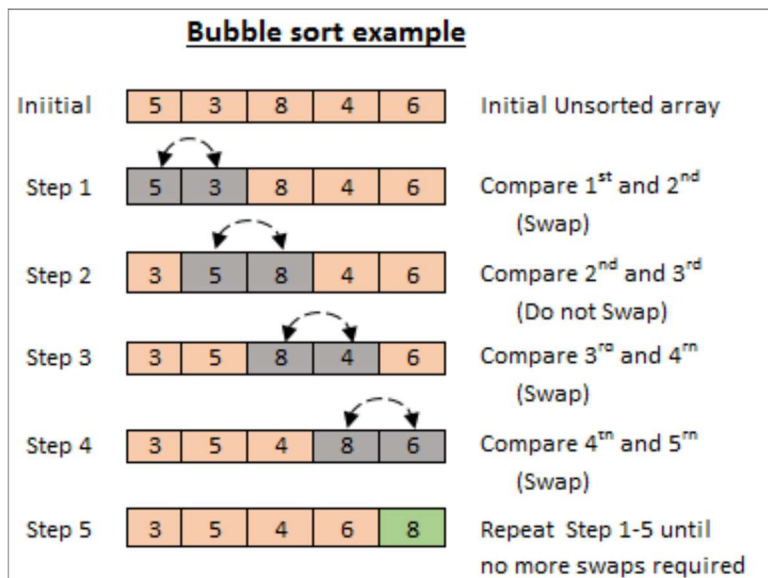
Exercice 1 : Permutation

Définir un algorithme Permuter qui prend en paramètre un tableau ainsi que deux entiers index1 et index2 et qui :

- permute dans tableau les éléments en positions index1 et index2
- retourne vrai si la permutation a pu se faire
- retourne faux sinon (car position(s) non accessible(s)).

Tester l'appel de cet algorithme, dans un programme principal

Exercice 2 : Tri à bulles

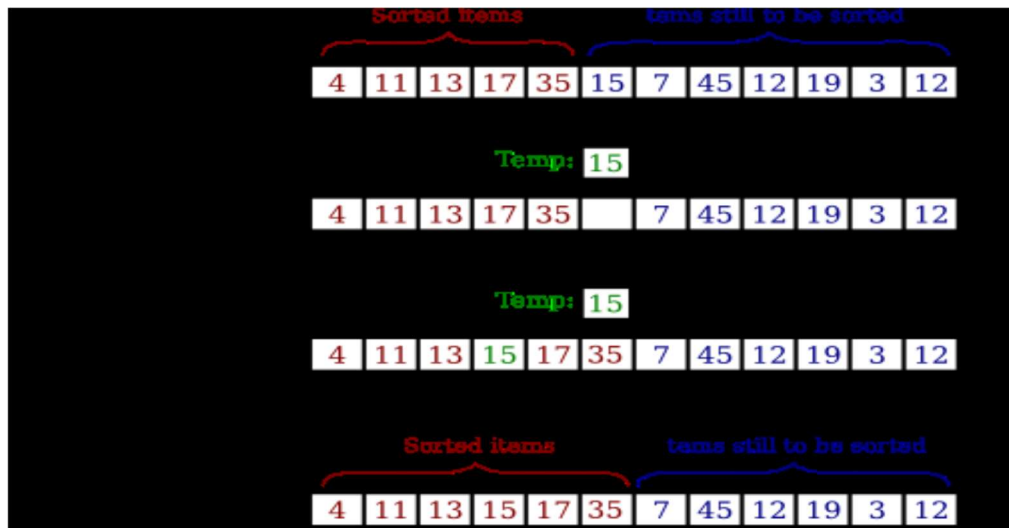


Le tri à bulles consiste à comparer répétitivement les éléments consécutifs d'un tableau, et à les permuter lorsqu'ils sont mal triés l'un par rapport à l'autre. Lors du premier parcours du tableau (étapes 1 à 5 ci-contre), le plus grand élément se retrouve nécessairement bien positionné.

On recommence alors un nouveau parcours, du premier à l'avant dernier élément. Et ainsi de suite jusqu'à ce que le tableau soit trié. Ce tri doit son nom au fait qu'il déplace rapidement les plus grands éléments en fin de tableau, comme des bulles d'air qui remonteraient rapidement à la surface d'un liquide.

- Ecrire un premier algorithme qui implémente un tri à bulles (tel qu'il a été expliqué sur le schéma) sur un tableau d'entiers, pour trier les valeurs dans l'ordre croissant.
- Si lors d'un parcours du tableau aucune permutation n'a été effectuée, qu'est-ce que cela signifie ?
- Si lors de la comparaison des deux derniers éléments à parcourir (donc pas nécessairement ceux en dernière et avant dernière positions du tableau) aucune permutation n'est à faire, que peut-on en déduire ?

Exercice 3 : Tri par insertion

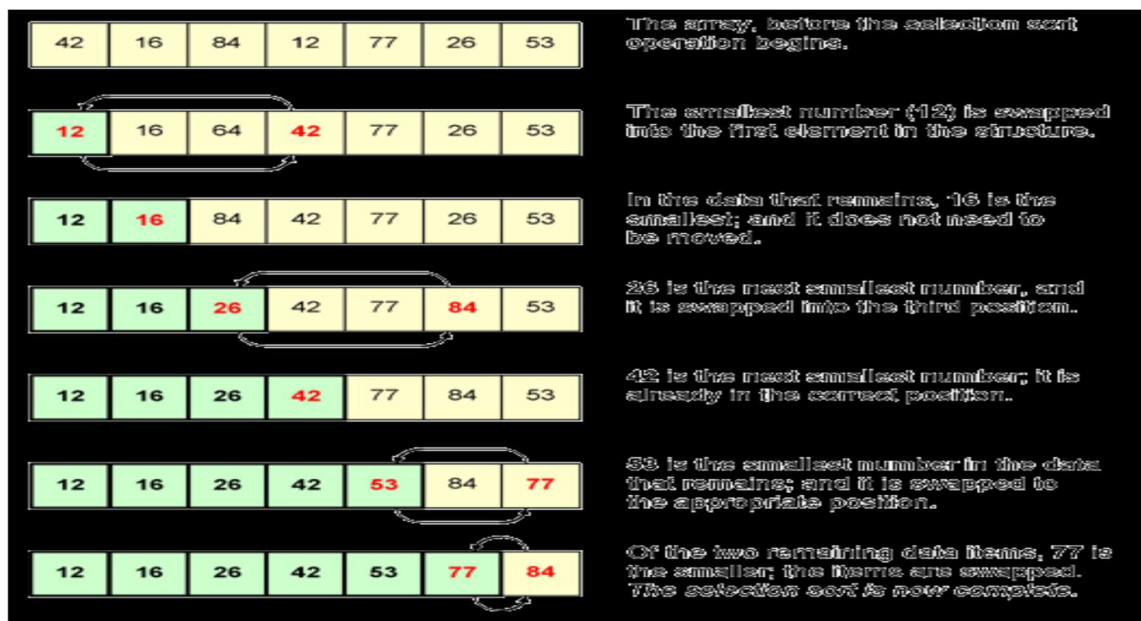


Le tri par insertion consiste à parcourir le tableau et pour chaque élément (à chaque index i) l'objectif est de l'insérer à la bonne place au regard des éléments précédemment visités. Ainsi, au moment où on considère le i ème élément, les éléments qui le précèdent sont déjà triés (zone en rouge ci-contre). L'insertion peut se faire de la façon suivante :

- en stockant temporairement la valeur du i ème élément,
- en parcourant le tableau vers la gauche depuis l'index i et en décalant d'un cran vers la droite les éléments plus grand que celui temporaire, puis
- en copiant l'élément temporaire à la position du dernier élément qui a pu être décalé ; cf. figure.

Ecrire l'algorithme correspondant au tri par insertion sur un tableau d'entiers, pour trier les valeurs dans l'ordre croissant.

Exercice 4 : Tri par sélection



Le tri par sélection consiste à parcourir le tableau, à trouver (sélectionner) le plus petit élément et à le permuter avec l'élément en première position. Ainsi, lors du premier parcours du tableau, le plus petit élément se retrouve nécessairement bien positionné. On recommence alors un nouveau parcours, du second au dernier élément. Et ainsi de suite jusqu'à ce que le tableau soit trié.

Ecrire l'algorithme correspondant au tri par sélection sur un tableau d'entiers, pour trier les valeurs dans l'ordre croissant.