Tris
TD3

Ce document est le TD adjoint au cours sur les tris.

Vous devez avoir lu le cours sur les différents tris afin de de réaliser ces exercices.

Swap

1) Écrivez une fonction qui inverse la position de deux éléments. Vous ne devez pas construire de nouveau tableau, mais uniquement modifier en place le tableau (en utilisant des variables temporaires). Par exemple, pour un tableau contenant [A B C D], si l'on inverse la position des éléments 0 et 1, le tableau doit devenir [B A C D]. *SwapEltTab(tab, len, pos1, pos2)*

1 Tri à bulles

Le tri à bulles vise à faire remonter tour à tour les plus grandes valeurs vers la fin du tableau. Si deux valeurs côté à côté sont dans le désordre, on les inverse, et on teste les suivantes jusqu'à la fin du tableau. Avec cette méthode, dès que l'on trouve la plus grande valeur du tableau, celle-ci avancera progressivement jusqu'à la fin. Avant d'atteindre la plus grande valeur, on peut également faire remonter les autres valeurs plus grandes. Cependant, à chaque fois que l'on parcourt intégralement le tableau de gauche à droite, on est sûr que la valeur la plus à droite est à sa place finale. Ainsi, on réduit la zone à parcourir à chaque tour : une fois la plus grande valeur du tableau trouvée à la fin du premier tour et placée à la dernière case, on n'a plus besoin de chercher dans cette case (l'élément étant maintenant à sa place).

2) Écrivez une procédure de tri respectant l'algorithme du tri à bulles (ou *bubble sort* en anglais). Vous devez modifier la position des éléments dans le tableau créer de tableau supplémentaire. *BubbleSort(tab, len)*

2 Tri par sélection

Le tri par sélection est le tri le plus évident humainement parlant : on cherche le plus grand élément dans le tableau en parcourant toutes les cases, puis, on échange la place de cet élément avec le tout dernier du tableau (pour le placer à sa place définitive). On recommence ce traitement (sans lire la dernière case à chaque fois) jusqu'à avoir ordonné tous les éléments.

3) Écrivez une procédure de tri respectant l'algorithme du tri par sélection (ou *selection sort* en anglais). Vous devez modifier la position des éléments dans le tableau sans créer de tableau supplémentaire. *SelectionSort(tab, len)*

3 Tri par insertion

Le tri par insertion considère que le tableau donné en paramètre n'est pas trié, et seuls les éléments qu'il a manipulé successivement le sont. Ainsi, chaque élément du tableau est comparé à tous ceux déjà triés, et on le place là où il devrait être. Pour le premier élément, celui-ci est considéré comme déjà trié, on peut donc passer au deuxième. Le deuxième est comparé avec l'unique élément déjà trié : on échange leurs deux places si nécessaire. Le troisième élément est comparé au plus grand des deux éléments triés, s'il est plus grand ou égal, il reste à sa place, sinon on le décale vers la gauche d'un cran, et on le compare à l'élément suivant (et ainsi de suite). Chaque élément du tableau est donc *inséré* à sa place parmi les éléments considérés comme triés.

4) Écrivez une procédure de tri respectant l'algorithme du tri par insertion (ou *insertion sort* en anglais). Vous devez modifier la position des éléments dans le tableau créer de tableau supplémentaire. *InsertionSort(tab, len)*

Ce document et ses illustrations ont été réalisés par Fabrice BOISSIER en octobre 2022.
La plupart des exercices sont inspirés du cahier d'algo de Nathalie "Junior" BOUQUET et Christophe "Krisboul" BOULLAY.
(dernière mise à jour octobre 2023)