

Travaux dirigés et travaux pratiques sur les Tris

Exercice 1 : Tri à bulles

L'algorithme de tri à bulles parcourt la liste, et compare les couples d'éléments successifs. Lorsque deux éléments successifs ne sont pas dans l'ordre croissant, ils sont échangés. Après chaque parcours complet de la liste, l'algorithme recommence l'opération. Lorsqu'aucun échange n'a lieu pendant un parcours, cela signifie que la liste est triée : l'algorithme peut s'arrêter.

- 1) (TD) Triez le tableau suivant en donnant toutes les étapes intermédiaires à l'aide du tri à bulles.

3	5	2	0	1	4
---	---	---	---	---	---

- 2) (TD) Donnez en pseudo-code, l'implémentation de la fonction `tri_bulles(tableau)` qui prend en paramètre un tableau et le tri à l'aide de l'algorithme de tri à bulles.
- 3) (TD) Donnez la complexité de cet algorithme.
- 4) (TP) Implémentez en Java la fonction `void tri_bulles(int tableau[])` qui ordonne les éléments du tableau suivant le tri à bulle

Exercice 2 : Tri par sélection

Il consiste en la recherche du plus grand élément (ou le plus petit) dans le tableau que l'on va replacer à sa position finale c'est-à-dire en dernière position (ou en première), puis on recherche le second plus grand élément (ou le second plus petit) que l'on va replacer également à sa position finale c'est-à-dire en avant-dernière position (ou en seconde), etc., jusqu'à ce que le tableau soit entièrement trié.

- 1) (TD) Triez le tableau suivant en donnant toutes les étapes intermédiaires à l'aide du tri par sélection.

3	5	2	0	1	4
---	---	---	---	---	---

- 2) (TD) Donnez en pseudo-code, l'implémentation de la fonction `tri_bulle(tableau)` qui prend en paramètre un tableau et le tri à l'aide de l'algorithme de tri par sélection.
- 3) (TD) Donnez la complexité de cet algorithme.
- 4) (TP) Implémentez en Java la fonction `void tri_bulle(int tableau[])` qui ordonne les éléments du tableau suivant le tri par sélection.

Exercice 3 : Tri par insertion

Le tri par insertion consiste à trier la liste au fur et à mesure que l'on ajoute un élément. Pour ce faire, on parcourt la liste en sélectionnant les éléments dans l'ordre. Pour chaque élément sélectionné, on le compare avec les éléments précédents qui ont déjà été ordonnés jusqu'à trouver sa place. Il ne reste plus qu'à décaler les éléments du tableau pour insérer l'élément considéré à sa place dans la partie déjà triée.

- 1) (TD) Triez le tableau suivant en donnant toutes les étapes intermédiaires à l'aide du tri par insertions.

3	5	2	0	1	4
---	---	---	---	---	---

- 2) (TD) Donnez en pseudo-code, l'implémentation de la fonction `tri_bulles(tableau)` qui prend en paramètre un tableau et le tri à l'aide de l'algorithme de tri par insertions.

- 3) (TD) Donnez la complexité de cet algorithme.
- 4) (TP) Implémentez en Java la fonction `void tri_bulles(int tableau[])` qui ordonne les éléments du tableau suivant le tri par insertions.

Exercice 4 : Tri rapide

La méthode consiste à placer un élément du tableau (appelé pivot) à sa place définitive, en permutant tous les éléments de telle sorte que tous ceux qui lui sont inférieurs soient à sa gauche et que tous ceux qui lui sont supérieurs soient à sa droite. Cette opération s'appelle le partitionnement. Pour chacun des sous-tableaux situés à droite et à gauche du pivot, on définit un nouveau pivot et on répète l'opération de partitionnement. Ce processus est répété récursivement, jusqu'à ce que l'ensemble des éléments soit trié.

Généralement, on choisit comme pivot le premier élément de la liste.

- 1) (TD) Triez le tableau suivant en donnant toutes les étapes intermédiaires à l'aide du tri rapide.

3	5	2	0	1	4
---	---	---	---	---	---

- 2) (TD) Donnez en pseudo-code, l'implémentation de la fonction `tri_bulles(tableau)` qui prend en paramètre un tableau et le tri à l'aide de l'algorithme de tri rapide.
- 3) (TD) Donnez la complexité de cet algorithme.
- 4) (TP) Implémentez en Java la fonction `void tri_bulles(int tableau[])` qui ordonne les éléments du tableau suivant le tri rapide.

Exercice 5 : Tri fusion

L'algorithme consiste à effectuer de manière récursive l'opération suivant :

- On découpe en deux parties à peu près égales les données à trier
- On trie les données de chaque partie
- On fusionne les deux parties

La récursivité s'arrête à un moment car on finit par arriver à des listes de 1 élément et alors le tri est trivial.

- 1) (TD) Triez le tableau suivant en donnant toutes les étapes intermédiaires à l'aide du tri fusion.

3	5	2	0	1	4
---	---	---	---	---	---

- 2) (TD) Donnez en pseudo-code, l'implémentation de la fonction `tri_bulles(tableau)` qui prend en paramètre un tableau et le tri à l'aide de l'algorithme de tri fusion.
- 3) (TD) Donnez la complexité de cet algorithme.
- 4) (TP) Implémentez en Java la fonction `void tri_bulles(int tableau[])` qui ordonne les éléments du tableau suivant le tri fusion.

Références

Les algorithmes présentés dans ce TD/TP sont tirés des pages de wikipedia consacrées aux tris.