

4TPM205U: Algorithmique des tableaux: feuille 7

Algorithmes de tri (1)

À traiter avant la feuille 6 en 2019-2020

Travaux dirigés

Exercice 1 – Tri par sélection : algorithme et analyse

Le principe du tri par sélection d'un tableau t de n éléments est le suivant :

- pour chaque valeur de i entre 0 et $n-2$, on parcourt le tableau entre l'indice i et l'indice $n-1$.
- à chaque parcours on cherche l'élément minimal et on l'échange ensuite avec l'élément d'indice i .

Exemple : soit $t = [7, 3, 5, 9, -1, 2]$ le tableau à trier. Les trois premières étapes sont :

- $i = 0$: on parcourt le tableau entre les indices 0 et 5 , on trouve le minimum à l'indice 4 : $t[4]$ est échangé avec $t[0]$ et on obtient $t = [-1, 3, 5, 9, 7, 2]$.
- $i = 1$: on parcourt le tableau entre les indices 1 et 5 , on trouve le minimum à l'indice 5 , on échange et on obtient $t = [-1, 2, 5, 9, 7, 3]$.
- $i = 2$: on parcourt le tableau entre les indices 2 et 5 , on trouve le minimum à l'indice 5 , on échange et on obtient $t = [-1, 2, 3, 9, 7, 5]$.

1. Appliquer le tri par sélection au tableau $t=[7, 5, -3, 4]$ en détaillant toutes les étapes.
2. Pour obtenir la fonction qui trie par sélection :
 - a. écrire une fonction `indiceMin`, qui renvoie l'indice d'un élément minimal d'un tableau, entre les indices i et $n-1$ inclus. Quels seront les paramètres de cette fonction ?
 - b. écrire une fonction `echanger` qui prend comme paramètres un tableau d'entiers et les indices (valides) de deux cases dont on veut échanger le contenu.
 - c. écrire une fonction `triSelection(t, n)` qui appelle `indiceMin` et `echanger` pour effectuer le tri du tableau t contenant n nombres.
3. Si on applique le `triSelection` à un tableau de n éléments, et si on arrête l'exécution de l'algorithme après k itérations ($1 \leq k < n$), combien d'éléments se trouveront à leur place définitive (et ne seront plus considérés si on reprend l'exécution) ?
4. Donner le nombre de comparaisons nécessaires entre éléments du tableau pour effectuer le tri par sélection d'un tableau de taille n dans le pire des cas (quel est-il ?), puis dans le meilleur des cas (quel est-il ?).
5. Donner le nombre d'affectations à une case du tableau nécessaires pour effectuer le tri par sélection d'un tableau de taille n dans le pire des cas (quel est-il ?), puis dans le meilleur des cas (quel est-il ?).
6. En fonction du nombre n d'éléments à trier, quel est l'ordre de grandeur de la complexité en temps du tri par sélection ?
7. Combien de fois au maximum déplace-t-on un élément du tableau ?
8. Si le tableau contient deux occurrences du même élément, est-ce qu'après le tri l'ordre initial des deux occurrences entre elles est encore respecté ?

Exercice 2 – Tri par insertion : algorithme et analyse

Le principe du tri par insertion d'un tableau t de n éléments est le suivant :

- Pour chaque valeur de i entre 1 et $n - 1$, on insère l'élément d'indice i à sa place dans la suite (déjà triée) des éléments d'indice inférieur à i .

Exemple : soit $t = [7, 3, 5, 9, -1, 2]$ le tableau à trier. Les trois premières étapes sont :

- $i = 1$: on insère 3 dans la suite $[7]$ et on obtient $t = [3, 7, 5, 9, -1, 2]$.
- $i = 2$: on insère 5 dans la suite $[3, 7]$ et on obtient $t = [3, 5, 7, 9, -1, 2]$.
- $i = 3$: on insère 9 dans la suite $[3, 5, 7]$: t ne change pas.

1. Appliquer le tri par insertion au tableau $t = [7, 5, -3, 4]$ en détaillant toutes les étapes.
2. En supposant que les éléments d'indice inférieur à i d'un tableau t sont déjà triés, quelles instructions permettent d'insérer $t[i]$ à sa place parmi les premiers $i+1$ éléments de t ?
3. Écrire une fonction `triInsertion(t, n)` qui met en œuvre l'algorithme de tri par insertion sur un tableau t contenant n nombres.
4. Si on applique le `triInsertion` à un tableau de n éléments, et si on arrête l'exécution de l'algorithme après k itérations ($1 \leq k < n$), combien d'éléments se trouveront à leur place définitive (et ne seront plus considérés si on reprend l'exécution) ?
5. Donner le nombre de comparaisons nécessaires entre éléments du tableau pour effectuer le tri par insertion d'un tableau de taille n dans le pire des cas (quel est-il ?). Puis dans le meilleur des cas (quel est-il ?).
6. Donner le nombre d'affectations à une case du tableau nécessaires pour effectuer le tri par insertion d'un tableau de taille n dans le pire des cas (quel est-il ?), puis dans le meilleur des cas (quel est-il ?).
7. En fonction du nombre n d'éléments à trier, quel est l'ordre de grandeur de la complexité en temps du tri par insertion ?
8. Combien de fois au maximum déplace-t-on un élément du tableau ?
9. Si le tableau contient deux occurrences du même élément, est-ce que après le tri l'ordre initial des deux occurrences entre elles est encore respecté ?