

## TD N° 5 : Les Algorithmes de Tri

### Exercice 1

Ecrire un programme qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri par sélection** du minimum. Le principe consiste à rechercher la position du minimum du tableau, à échanger ce minimum avec la première case du tableau courant, et à répéter le même travail à partir de l'indice suivant.

Le principe du tri par sélection est illustré dans l'exemple suivant :

On considère le tableau :

5	8	11	2	6	1	3	7	10	9
---	---	----	---	---	---	---	---	----	---

**1<sup>ère</sup> itération**, on considère le tableau à partir de l'indice 0, on repère le min et on le permute avec la 1<sup>ère</sup> case

5	8	11	2	6	1	3	7	10	9
---	---	----	---	---	---	---	---	----	---

1	8	11	2	6	5	3	7	10	9
---	---	----	---	---	---	---	---	----	---

**2<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 1, et on refait le même traitement

1	8	11	2	6	5	3	7	10	9
---	---	----	---	---	---	---	---	----	---

1	2	11	8	6	5	3	7	10	9
---	---	----	---	---	---	---	---	----	---

**3<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 2, et on refait le même traitement

1	2	11	8	6	5	3	7	10	9
---	---	----	---	---	---	---	---	----	---

1	2	3	8	6	5	11	7	10	9
---	---	---	---	---	---	----	---	----	---

**4<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 3, et on refait le même traitement

1	2	3	8	6	5	11	7	10	9
---	---	---	---	---	---	----	---	----	---

1	2	3	5	6	8	11	7	10	9
---	---	---	---	---	---	----	---	----	---

**5<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 4, et on refait le même traitement

1	2	3	5	6	8	11	7	10	9
---	---	---	---	---	---	----	---	----	---

(pas de permutation le min est déjà à la première position !!)

1	2	3	5	6	8	11	7	10	9
---	---	---	---	---	---	----	---	----	---

**6<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 5, et on refait le même traitement

1	2	3	5	6	8	11	7	10	9
---	---	---	---	---	---	----	---	----	---

1	2	3	5	6	7	11	8	10	9
---	---	---	---	---	---	----	---	----	---

**7<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 6, et on refait le même traitement

1	2	3	5	6	7	11	8	10	9
---	---	---	---	---	---	----	---	----	---

1	2	3	5	6	7	8	11	10	9
---	---	---	---	---	---	---	----	----	---

**8<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 7, et on refait le même traitement

1	2	3	5	6	7	8	11	10	9
---	---	---	---	---	---	---	----	----	---

1	2	3	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	----	----

**9<sup>ème</sup> itération**, on considère le tableau à partir de l'indice 8, et on refait le même traitement

1	2	3	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	----	----

(pas de permutation le min est déjà à la première position !!)

1	2	3	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	----	----

Il n'est pas nécessaire de faire une itération pour traiter la dernière case, car elle contient certainement la plus grande valeur.

**Remarque :** En suivant le même principe on peut faire un tri par sélection du max, il s'agit de repérer la plus grande valeur et de la permuter à chaque étape avec la dernière case du tableau courant, répéter alors le même processus à partir de l'indice précédent.

## Exercice 2

Ecrire un programme qui permet de trier un tableau en effectuant un **tri par Bulles**. Le principe est le suivant : comparer les valeurs successives du tableau deux par deux, et effectuer une permutation à chaque fois qu'une valeur est supérieure à la suivante. Répéter ce traitement jusqu'à ce qu'il n'y ait plus de permutations possibles.

Le fonctionnement de l'algorithme sur l'exemple précédent est illustré comme suit :

Tableau initial

5	8	11	2	6	1	3	7	10	9
---	---	----	---	---	---	---	---	----	---

**1<sup>ère</sup> itération :**

5	8	11	2	6	1	3	7	10	9
5	8	11	2	6	1	3	7	10	9
5	8	2	11	6	1	3	7	10	9
5	8	2	6	11	1	3	7	10	9
5	8	2	6	1	11	3	7	10	9
5	8	2	6	1	3	11	7	10	9
5	8	2	6	1	3	7	11	10	9
5	8	2	6	1	3	7	10	11	9
5	8	2	6	1	3	7	10	9	11

indice de la dernière  
permutation effectuée

**2<sup>ème</sup> itération :**

5	8	2	6	1	3	7	10	9	11
5	2	8	6	1	3	7	10	9	11
5	2	6	8	1	3	7	10	9	11
5	2	6	1	8	3	7	10	9	11
5	2	6	1	3	8	7	10	9	11
5	2	6	1	3	7	8	10	9	11
5	2	6	1	3	7	8	10	9	11
5	2	6	1	3	7	8	9	10	11

indice de la dernière permutation

**3<sup>ème</sup> itération :**

2	5	6	1	3	7	8	9	10	11
2	5	6	1	3	7	8	9	10	11
2	5	1	6	3	7	8	9	10	11
2	5	1	3	6	7	8	9	10	11
2	5	1	3	6	7	8	9	10	11
2	5	1	3	6	7	8	9	10	11
2	5	1	3	6	7	8	9	10	11

indice de la dernière permutation

**4<sup>ème</sup> itération :**

2	5	1	3	6	7	8	9	10	11
2	1	5	3	6	7	8	9	10	11
2	1	3	5	6	7	8	9	10	11

indice de la dernière permutation

**5<sup>ème</sup> itération :**

1	2	3	5	6	7	8	9	10	11
1	2	3	5	6	7	8	9	10	11

indice de la dernière permutation

Il n'est pas nécessaire de faire une autre itération car la première du tableau contient forcément la plus petite valeur, puisque le tri bulle a propagé à chaque fois les valeurs max vers la fin du tableau.

**Remarque :** ce tri permet de gagner en terme de nombre d'itérations, il suffit alors de repérer l'indice où a eu lieu la dernière permutation et à l'itération suivante s'arrêter à cet indice, puisque tous les indices supérieurs sont forcément triés (puisque'il n'y a pas eu de permutation).

### Exercice 3

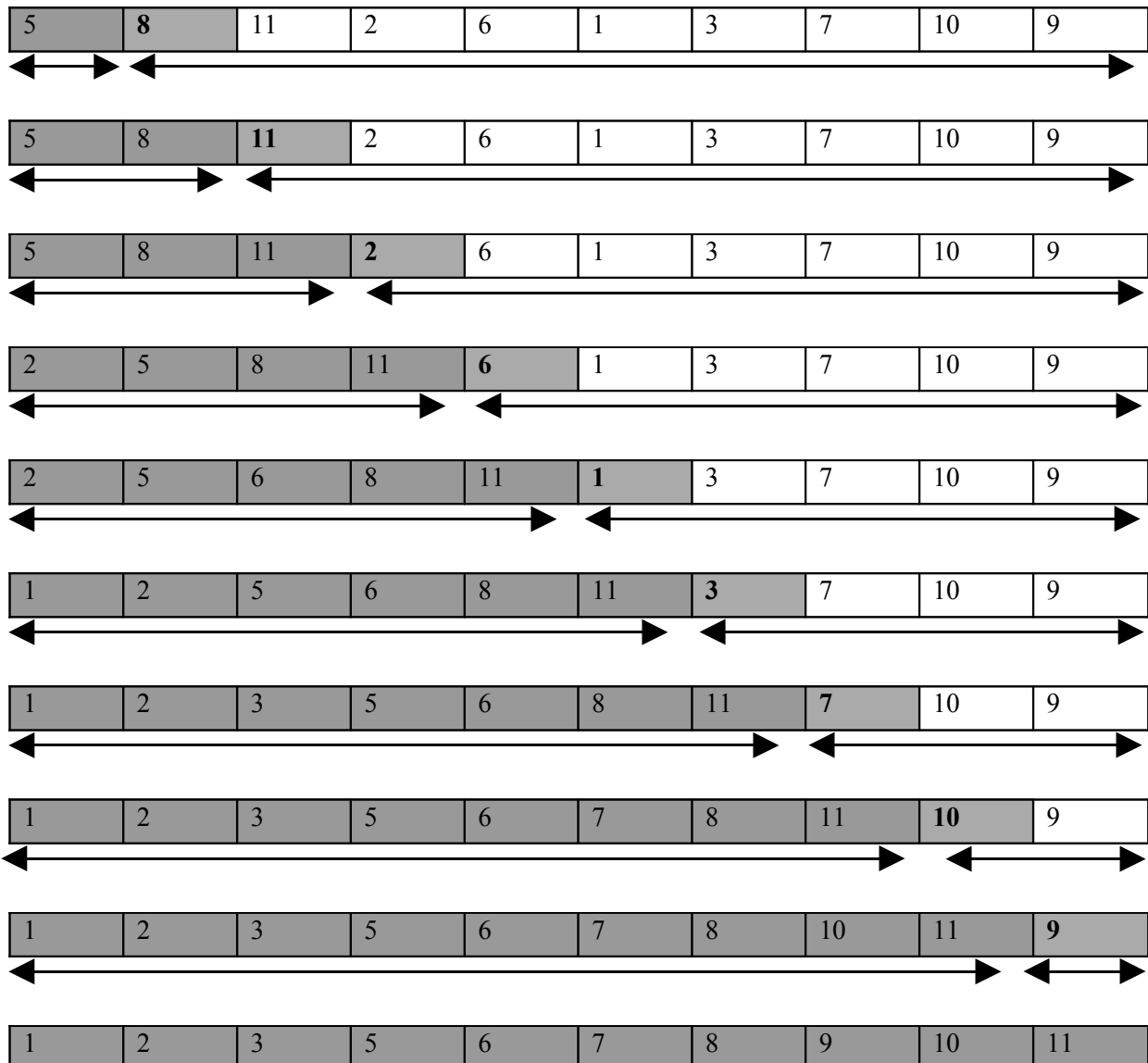
Ecrire un programme qui permet de trier un tableau en utilisant la méthode de **tri par insertion**. Le tri par insertion est basé sur les principes suivants : dans le tableau à insérer, on suppose qu'une partie a été triée et qu'il reste à trier l'autre partie. La partie triée est appelée 'séquence destination' et la partie qui reste à trier est appelée 'séquence source'. A chaque étape, un élément de la séquence source est inséré dans la séquence destination de manière à ce que celle-ci reste triée. Le processus est répété jusqu'à la fin de la séquence source.

$a_0, a_1, a_2, a_3, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n$

séquence destination      séquence source

( triée)

( non triée)



### Exercices d'entraînement

#### Exercice 4

Écrire un programme permettant de saisir puis de trier en ordre croissant un tableau de caractères.