

TD 3 : algorithmes de tri

Soit un tableau T de taille N où chaque cellule c_i avec $1 \leq i \leq N$ contient un nombre. Trier T par ordre croissant consiste à réordonner les valeurs des cellules de manière à ce que :

$$\forall 1 \leq i \leq N \text{ et } 1 \leq j \leq N : c_i \leq c_j$$

Pour passer d'un tableau non trié à un tableau trié par ordre croissant, il existe plusieurs méthodes de tri. L'objectif de ce projet est de programmer deux de ces méthodes (des tris naïfs assez peu efficaces au final 😊) pour pouvoir ensuite comparer leur efficacité.

Exemple

Pour illustrer les deux méthodes de tri décrites ci-dessous, nous prendrons le tableau T comprenant 5 éléments et dont les éléments sont :

8	5	7	1	2
1	2	3	4	5

Exercice 1 : Fonction d'insertion d'un élément dans un tableau

Pour réaliser les deux algorithmes de tri décrits ci-après, vous aurez besoin d'une fonction d'insertion d'un élément. Considérons que cet élément se trouve à la position i et doit être inséré à la position j (avec $j < i$).

Pour insérer $T[i]$ à la position j , il faut :

1. Sauvegarder la valeur de $T[i]$ dans une variable x ;
2. Décaler d'une position tous les éléments compris entre la position j et la position $i-1$;
3. Mettre la valeur de x à la position j .

Programmez une fonction Octave qui réalise l'insertion d'un élément à partir d'un tableau de nombre entiers, d'une position d'un nombre à insérer et d'un indice de position d'arrivée.

Exercice 2 : Fonction de Tri par sélection

Le tri par sélection consiste à chercher le plus petit élément du tableau et à le placer en première position. Une fois le plus petit élément positionné en première position, on recommence la même opération en commençant à l'indice 2 : on cherche le plus petit élément compris dans le tableau entre la position 2 et la fin du tableau, puis on place cet élément à la position 2. Puis on recommence ainsi en partant de l'élément 3, 4 et ainsi de suite jusqu'au bout du tableau.

Soit le tableau de départ suivant	<table><tr><td>8</td><td>5</td><td>7</td><td>1</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	8	5	7	1	2	1	2	3	4	5
8	5	7	1	2							
1	2	3	4	5							
Le plus petit élément du tableau est 1. On l'insère à la première position du tableau	<table><tr><td>1</td><td>8</td><td>5</td><td>7</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	8	5	7	2	1	2	3	4	5
1	8	5	7	2							
1	2	3	4	5							
On recommence en partant de l'indice 2. Entre l'élément 2 et la fin du tableau, le plus petit élément est 2. On l'insère donc en seconde position.	<table><tr><td>1</td><td>2</td><td>8</td><td>5</td><td>7</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	8	5	7	1	2	3	4	5
1	2	8	5	7							
1	2	3	4	5							

On recommence en partant de l'indice 3. Entre l'élément 3 et la fin du tableau, le plus petit élément est 5. On l'insère donc en troisième position.	<table><tr><td>1</td><td>2</td><td>5</td><td>8</td><td>7</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	5	8	7	1	2	3	4	5
1	2	5	8	7							
1	2	3	4	5							
On recommence en partant de l'indice 4. Entre l'élément 4 et la fin du tableau, le plus petit élément est 7. On l'insère donc en quatrième position.	<table><tr><td>1</td><td>2</td><td>5</td><td>7</td><td>8</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	5	7	8	1	2	3	4	5
1	2	5	7	8							
1	2	3	4	5							

Programmez une fonction GNU Octave qui réalise le tri par sélection d'un tableau.

Exercice 3 : Fonction de Tri par insertion

Le tri par insertion consiste à classer les deux premiers éléments du tableau. Une fois que les deux premiers sont ordonnés, on prend l'élément qui suit et on le classe à son tour dans ce qui a déjà été classé. Pour chaque élément i du tableau, on sait que les éléments de 1 à $i-1$ sont déjà classés. On va chercher la position j parmi les $i-1$ premiers éléments de manière à ce que $T[i] < T[j]$. On insère alors $T[i]$ à la position j .

Soit le tableau de départ suivant	<table><tr><td>8</td><td>5</td><td>7</td><td>1</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	8	5	7	1	2	1	2	3	4	5
8	5	7	1	2							
1	2	3	4	5							
On commence par classer les 2 premiers éléments	<table><tr><td>5</td><td>8</td><td>7</td><td>1</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	5	8	7	1	2	1	2	3	4	5
5	8	7	1	2							
1	2	3	4	5							
Puis on prend l'élément à la position $i=3$ ($T[3] = 7$), on cherche la position j dans le sous tableau déjà trié ($[5 ; 8]$) : $j=2$, $x=T[3]=7$, on décale tous les éléments d'une case entre j et $i-1$ soit ici entre 2 et 2. Donc seul 8 est décalé d'une case. Enfin on insère x à la position j .	<table><tr><td>5</td><td>7</td><td>8</td><td>1</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	5	7	8	1	2	1	2	3	4	5
5	7	8	1	2							
1	2	3	4	5							
On prend ensuite l'élément à la position $i=4$ ($T[4] = 1$), on cherche la position j dans le sous tableau déjà trié ($[5 ; 7 ; 8]$) : $j=1$, $x=T[4]=1$, on décale tous les éléments d'une case entre j et $i-1$ soit ici entre 1 et 3. Donc 5, 7 et 8 sont décalés d'une case. Enfin on insère x à la position j .	<table><tr><td>1</td><td>5</td><td>7</td><td>8</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	5	7	8	2	1	2	3	4	5
1	5	7	8	2							
1	2	3	4	5							
Enfin, on prend l'élément à la position $i=5$ ($T[5] = 2$), on cherche la position j dans le sous tableau déjà trié ($[1 ; 5 ; 7 ; 8]$) : $j=2$, $x=T[5] = 2$, on décale tous les éléments d'une case entre j et $i-1$ soit ici entre 2 et 4. Donc 5, 7 et 8 sont décalés d'une case. Enfin on insère x à la position j .	<table><tr><td>1</td><td>2</td><td>5</td><td>7</td><td>8</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	5	7	8	1	2	3	4	5
1	2	5	7	8							
1	2	3	4	5							

Programmez une fonction GNU Octave qui réalise le tri par insertion d'un tableau.