NF16 - Tri - 1

$$\sum_{k=1}^{m} k = \frac{m(m+1)}{2}$$

Exercice 1

Tri par sélection : - Chercher le min de T[1..m]
- Le mettre en T[1] via échange
- Chercher le min de T[2..m]
- Le mettre en T[2] via échange

```
Tri_Selection(T : tableau ; m : entier)
    i, j, pos_min : entier
    Pour i de 1 à m-1
        pos_min := i
        Pour j de i+1 à m
            Si T[j] < T[pos_min]
                pos_min := j
        Swap(T, i, pos_min)
```

38 27 43 3 9 82 10
3, 27 43 38 9 82 10
9, 43 38 27 82 10
10, 38 27 82 43
27, 38 82 43
38, 82 43
43 82

Complexité : On compte le nombre d'itérations de j :

$$\sum_{i=1}^{m-1} (m-i) = (m-1) + (m-2) + \dots + 1$$
$$= \sum_{k=1}^{m-1} k = \frac{m(m-1)}{2} \Rightarrow O(m^2)$$

NF16 - Tri - 2

Tri à bulle : - Parcours le tableau en commençant par la fin de n à 1
- échange un élément avec son prédecesseur si ils ne sont pas dans le bon ordre
- Parcours le tableau de n à 2 ....

```
Tri_bulle (T : tableau ; n : entier)
    i, j : entier
    Pour i de 1 à n-1
        tab_trié := vrai
        Pour j de n à i+1
            Si T[j] < T[j-1]
                swap(T, j, j-1)
                tab_trié := faux
        Si tab_trié
            FIN
```

```
      38 27 43 3 9 82 10
i=1                 10 82
                  9 10
                3 9
             3 43
          3 27
       3 38
      ----
      3 38 27 43 9 10 82
```

Complexité : $\Theta(n^2)$

Amélioration : On s'arrête lorsque pour une itération de i, aucune inversion n'est réalisée $\Omega(n)$ (meilleur cas)

NF16 - Tri - 3

Tri par insertion : - Les i-1 premiers éléments du tableau sont triés (récursif)
- Inséré le i^e élément à la bonne position pour avoir un tableau avec les i premiers éléments triés

Tri_insertion (T : tableau ; i : entier)

38, 27, 43, 3, 9, 82, 10,
[3, 9, 10, 27, 38, 43, 82,
↳ 3 | 9 | 27 | 38 | 43 | 82
↳ 3 | 9 | 27 | 38 | 43 |
↳ 3 | 27 | 38 | 43 |
↳ 27 | 38 | 43 |
↳ 27 | 38 |
38 |

```
j : entier
Si i > 0
  | Tri_insertion (T, i-1)
  | j := i
  | TantQue (j > 1 et T[j-1] > T[j])
  |   | Swap(T, j-1, j)
  |   | j := j-1
```

Complexité pire des cas : Le tableau est initialement trié dans l'ordre décroissant. Soit $C(n)$ le nombre d'exécutions de la boucle TantQue dans le pire des cas pour un tableau de taille n.

$$C(n) = \begin{cases} 0 \text{ si } n = 0 \\ (n-1) + \underbrace{C(n-1)}_{\text{Appel récursif}} \text{ si } n > 0 \end{cases}$$

$$C(n) = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$$

$$\Theta(n^2)$$

Complexité moyenne :

Une fois le tableau $T[1..i-1]$ trié, l'élément $T[i]$ doit être remonté jusqu'à sa position. On suppose équiprobabilité sur l'indice ~~entre~~ $T[i]$ doit être remonté. En moyenne, $T[i]$ est remonté à sa place en $\frac{n-1}{2}$ swap. On pose $C'(n)$

$$C'(n) = \begin{cases} 0 \text{ si } n = 0 \\ C'(n-1) + \frac{n-1}{2} \text{ si } n > 0 \end{cases}$$

$$C'(n) = \frac{n(n-1)}{4}$$

NF16 - Tri - 4

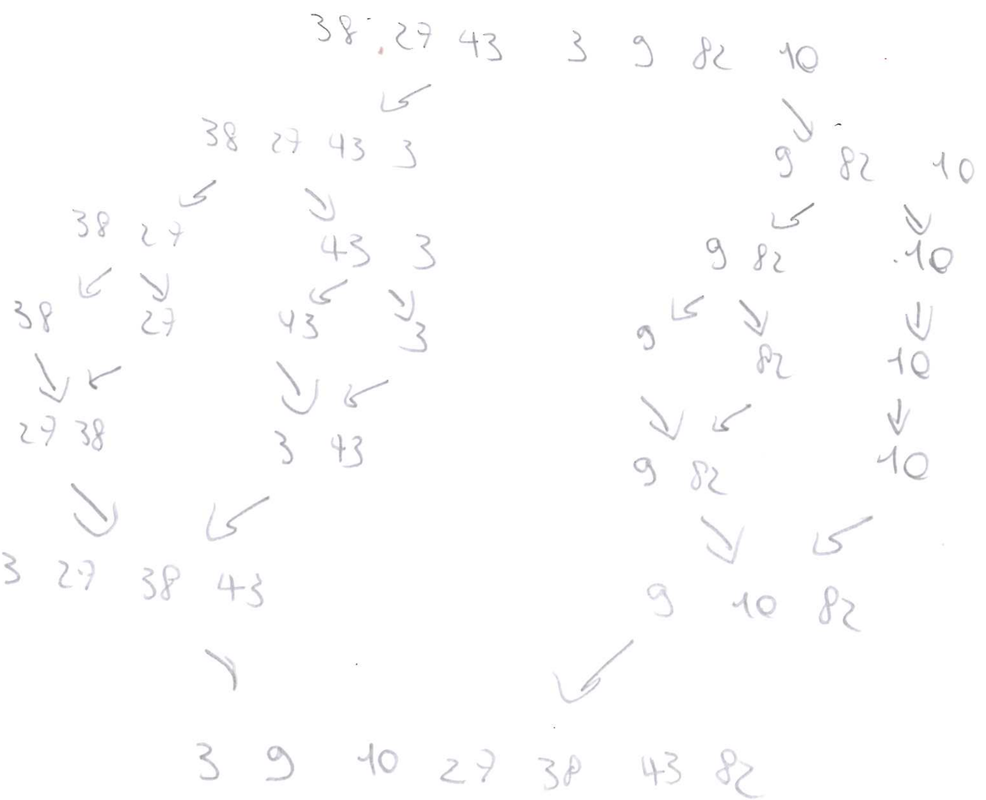
Exercice 2

Tri fusion : - Scinder le tableau $T[1..n]$ en 2 sous tableaux de taille $\frac{n}{2}$

- Trier les 2 sous tableaux
- Interclasser les 2 sous-tableaux

1) Tri_Fusion(T : tableau ; i, j : entier)

```
k : entier
Si i < j
    k := ⌊(i+j)/2⌋
    Tri-Fusion(T, i, k)
    Tri-Fusion(T, k+1, j)
    Interclassement(T, i, j, k)
```



NF16 - Tri - 5

```
Interclassement(T: tableau, i, j, k : entier)
    R : tableau
    iter, ind_g, ind_d : entier
    Pour iter de i à k
    |   R[iter] := T[iter]
    Pour iter de k+1 à j
    |   R[iter] := T[j + k+1 - iter]
    ind_g := i
    ind_d := j

    Pour iter de i à j
    |   Si R[ind_g] < R[ind_d]
    |   |   T[iter] := R[ind_g]
    |   |   ind_g := ind_g + 1
    |   Sinon
    |   |   T[iter] := R[ind_d]
    |   |   ind_d := ind_d - 1
```

R : Sentinelles

3 27 38 43 | 9 10 82

ind_g ind_d

Complexité : Soit n la taille du tableau.

On parcourt 2 fois le tableau : $\Theta(n)$

NF16 - Tri - 6

Complexité : $C(n) = \begin{cases} 0 \text{ si } n = 0 \\ 2 \times C\left(\lfloor \frac{n}{2} \rfloor\right) + 2n \text{ sinon} \end{cases}$

$\Pi k \; \forall k \overset{\text{entier}}{\geq} 0, \; C(2^k) = (k+1) \times 2^{k+1}$

<u>Pour $k=0$:</u> $C(2^0) = C(1) = 2 \times C(0) + 2 = 2 = (0+1) \times 2^{0+1}$ ✓

<u>Hérédité :</u> Au rang $k+1$:

$$C(2^{k+1}) = 2 \times C\left(\frac{2^{k+1}}{2}\right) + 2 \times 2^{k+1}$$
$$= 2 \times (k+1) \times 2^{k+1} + 2 \times 2^{k+1}$$
$$= (k+2) \times 2^{k+2} \quad ✓$$

<u>Pour $k = \log_2 n$:</u> $n = 2^k$

$$C(2^{\log_2 n}) = (\log_2(n)+1) \times 2^{\log_2(n)+1}$$
$$= (\log_2(n)+1) \times 2n \Rightarrow \Theta(n \log_2 n)$$

<u>Pour $n \neq 2^k$:</u> $\exists p$ tq $2^p \leq n \leq 2^{p+1}$

$C(2^p) \leq C(n) \leq C(2^{p+1})$ On $p+1 > \log_2(n) > p$

$(p+1) \times 2^{p+1} \leq C(n) \leq (p+2) \times 2^{(p+2)}$

$\log_2 n \times 2^{\log_2 n} \leq C(n) < (\log_2(n)+2) \times 2^{\log_2(n)+2}$

$n \log_2 n \leq C(n) < 4n(\log_2(n)+2)$

$$\Rightarrow \Theta(n \log_2 n)$$

NF16 - TD13

Exercice 3

1) n : nombre d'objets O_i

$T[1..n]$: Tableau contenant les objets $T[i] = O_i$ initialement

$c_i = (c_i^1, \dots, c_i^q)$: Clé associé à l'objet i

K : $0 \leq c_i^p \leq K$ $\forall i \in [1..n]$, $\forall p \in [1..q]$

→ Classement par ordre lexico. croissant

$n = 10$, $q = 3$, $K = 10$

$T[1..10] = \{O_1, \dots, O_{10}\}$

$c_1 = (2,1,4)$
$c_2 = (2,9,7)$
$c_3 = (0,3,3)$
$c_4 = (2,9,8)$
$c_5 = (2,1,4)$
$c_6 = (2,0,4)$
$c_7 = (9,9,9)$
$c_8 = (2,2,2)$
$c_9 = (1,3,5)$
$c_{10} = (9,9,9)$

Init $O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8, O_9, O_{10}$

It 1 $O_8, O_3, O_1, O_5, O_6, O_9, O_2, O_4, O_7, O_{10}$

It 2 $O_6, O_1, O_5, O_8, O_3, O_9, O_2, O_4, O_7, O_{10}$

It 3 $O_3\ O_9\ O_6\ O_1\ O_5\ O_8\ O_2\ O_4\ O_7\ O_{10}$

queue file ↓ tête file ↓

K files

	It 1	It 2	It 3
0		O_6	O_3
1		$O_5\ O_1$	O_9
2	O_8	O_8	$O_4\ O_2\ O_8\ O_5\ O_1, O_6$
3	O_3	$O_9\ O_3$	
4	$O_6\ O_5\ O_1$		
5	O_9		
6			
7	O_2		
8	O_4		
9	$O_{10}\ O_7$	$O_{10}\ O_7\ O_4\ O_2$	$O_{10}\ O_7$
~~10~~	~~O_{10}~~		

3) Complexité :

Pour $x = g$ à 1 ... → g itérations

Pour $i = 1$ à n ... → n it

Pour $k = 0$ à $K-1$
Tant que Est_vide (F_k) = Faux } $K+n$ itérations

$$\rightarrow \mathcal{O}\left(g \times (n + K + n)\right)$$

Si $n \gg K$: $\mathcal{O}(gn)$

2) Validité

Après chaque itération de la boucle principale, le tableau est trié par ordre lexico en prenant en compte les composantes $(c_i^r, \ldots, c_i^g)$ associée à chaque clé

Supposons cette prop vraie à l'issue de l'it $\overset{r=}{t}$
$r = t-1$ (it suivante) objets
Lors de l'it ~~t+1~~ : Prenons 2 ~~elem~~ : l'un O_u se trouve à l'indice a du tableau et l'autre O_v à l'indice $b > a$.

Si $c_u^{t-1} < c_v^{t-1}$: L'elem O_u est placé dans une file d'indice plus petit → il sera defilé en premier OK

$c_u^{t-1} = c_v^{t-1}$: L'elem O_u sera enfilé avant O_v sur la file c_u^{t-1}. Il sera defilé en premier OK

$c_u^{t-1} > c_v^{t-1}$: L'elem O_u est placé dans la une file d'indice plus grand → il sera defilé après O_v OK