

Exercice 1.

Par groupe de 4 élèves, effectuer les actions suivantes :

1. Dans un jeu de 32 cartes, regrouper les cartes d'une même couleur (cœur, carreau, trèfle, pique).
2. Récupérer chacun une couleur.
3. Mélanger chaque paquet.
4. Trier dans l'ordre croissant (l'as est la carte de plus grande valeur).
5. Exposer et échanger entre vous la manière de faire. Utiliser le cadre réponse ci-dessous.

6. Mélanger à nouveau les cartes et recommencer le tri en commençant par placer à gauche la carte de plus petite valeur. Est-ce la méthode utilisée précédemment ?

7. Rechercher sur Internet les méthodes de tri par insertion et de tri par sélection. Exposer ci-dessous le principe de ces méthodes. Est-ce que ces méthodes correspondent à celle que vous avez expérimentée aux questions (4) et (6) ?

Exercise 2.

Étant donné un tableau TAB de taille *Taillemax*, on cherche à trier ses éléments dans l'ordre croissant. Pour cela, il existe plusieurs méthodes ; dans ce cours, nous allons en décrire deux :

- Le tri par sélection
- Le tri fusion

1 Le tri par sélection

C'est la méthode de tri la plus intuitive, mais malheureusement elle est peu efficace en terme de complexité.

Le principe : on recherche le plus grand (ou petit) élément du tableau TAB et on l'échange avec le dernier (ou le premier) élément du tableau. On poursuit ces opérations jusqu'à ce que le tableau soit trié.

Un lien pour voir fonctionner le processus : http://lwh.free.fr/pages/algo/tri/tri_insertion.html

1.1 Le tri par sélection : algorithme

```
Triselection(A : tableau, Taillemax : entier)
  variable locale    i, pos : entier
  DÉBUT
    fin ← Taillemax
    TANT QUE (.....) FAIRE
      pos ← 1
      plusgrand ← A[1]
      Pour i variant de 2 à ..... faire
        Si A[i] > plusgrand alors
          plusgrand=A[i]
          pos=i
        Fin si
      Fin pour
      Si pos ≠ fin alors Echanger(A[pos], A[fin])
      fin ← fin-1
    FIN TANT QUE
    retourner(A)
  FIN
```

1.2 Le tri par sélection : un exemple

Trier les éléments du tableau ci-dessous en déroulant l'algorithme précédent à la main :

8	3	6	5	4
---	---	---	---	---

Étape 1

A =

8	3	6	5	4
---	---	---	---	---

fin=5
pos =1
plusgrand = A[1] = 8 et 8 reste le plus grand élément du tableau donc pos est encore égal à 1 à la fin de la boucle POUR ; on échange A[1] et A[5], soit 8 et 4 :

A =

4	3	6	5	8
---	---	---	---	---

Étape 2

A =

4	3	6	5	8
---	---	---	---	---

fin=4
pos =1
plusgrand = A[1] = 4 et 4 n'est pas le plus grand élément du tableau restant ; à la fin de la boucle POUR, plusgrand est égal à 6 et pos vaut 3 ; on échange A[3] et A[4], soit 6 et 5 :

A =

4	3	5	6	8
---	---	---	---	---

Étape 3

A =

4	3	5	6	8
---	---	---	---	---

fin=3
pos =1
plusgrand = A[1] = 4 et 4 n'est pas le plus grand élément du tableau restant ; à la fin de la boucle POUR, plusgrand est égal à 5 et pos vaut 3 ; comme les variables pos et fin sont égales, 5 reste à sa place :

A =

4	3	5	6	8
---	---	---	---	---

Étape 4

A =

4	3	5	6	8
---	---	---	---	---

fin=2
pos =1
plusgrand = A[1] = 4 et 4 reste le plus grand élément du tableau restant ; donc pos est encore égal à 1 à la fin de la boucle POUR ; on échange A[1] et A[2], soit 4 et 3 :

A =

3	4	5	6	8
---	---	---	---	---

Étape 5

A =

3	4	5	6	8
---	---	---	---	---

fin=1 C'est terminé le tableau A est trié

Un autre exemple avec animation : http://lwh.free.fr/pages/algo/tri/tri_selection.html

1.3 Le tri par sélection : complexité

1. Combien de comparaisons vont être effectuées dans tous les cas ?
2. Si le tableau contient 20 millions de valeurs, combien y aura-t-il de comparaisons dans tous les cas ?

ADMIS

La complexité de la méthode de tri par sélection d'un tableau contenant n éléments est $O(n^2)$.

2 Le tri fusion

Le premier algorithme de tri fusion a été écrit par Von Neumann l'EDVAC en 1945.

Cette méthode de tri utilise le principe de récursivité, c'est-à-dire que la méthode s'appelle elle-même.

- **Diviser** : on divise le tableau à trier en deux sous-tableaux de même taille (au "milieu").
- **Régner** :
 - Tout tableau de taille 1 est trié
 - On trie les deux sous-tableaux (récursivement) ; on s'arrête lorsque les sous-tableaux sont de taille 1.
- **Combiner** : on fusionne les deux sous tableaux triés en un tableau trié. C'est à cette étape que le tableau est trié.
- **Complexité** : $n \log_2(n)$

Un lien pour visualiser le processus : http://lwh.free.fr/pages/algo/tri/tri_fusion.html

2.1 Le tri fusion : algorithme(1)

```
Trifusion(A : tableau d'entiers, deb : entier, fin : entier)
  variable locale    m : entier
  DÉBUT
    SI (deb < fin) ALORS
      m ← E((deb+fin)/2)
      Trifusion(A,deb,m)
      Trifusion(A,m+1,fin)
      Fusion(A,deb,m,fin)
    FIN SI
  FIN
```

2.2 Le tri fusion : algorithme(2)

Les deux sous-tableaux sont triés ; on les fusionne en un seul.

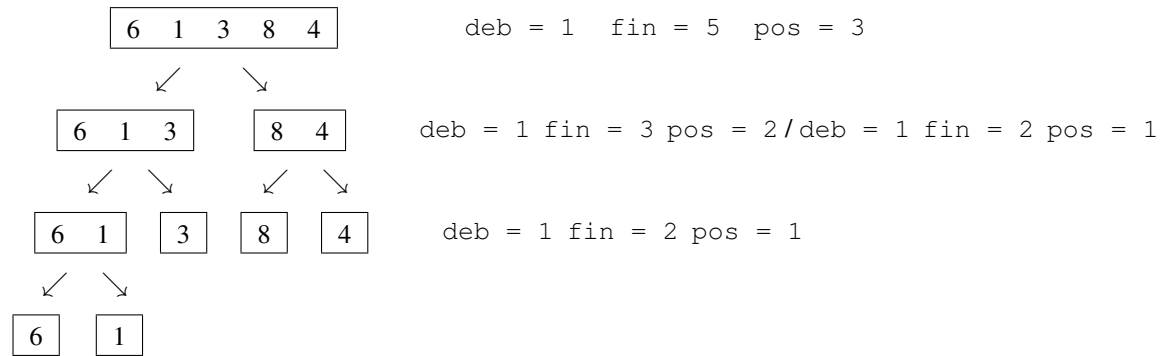
```
Fusion(A : tableau d'entiers, deb : entier, pivot : entier, fin : entier)
  variables locales  i,j,k : entier    B : tableau d'entiers
  DÉBUT
    i ← deb    k ← deb    j ← pivot+1
    TANT QUE (i ≤ ..... ) et (.....) FAIRE
      Si A[i] < A[j] alors
        B[k]=A[i]
        i ← i+1
      Sinon
        B[k]=A[j]
        j ← j+1
      Fin Si
      k ← k+1
    FIN TANT QUE
    On a atteint la fin du deuxième tableau, mais il reste des éléments du
premier tableau : on complète le tableau B avec ces éléments
    TANT QUE (i ≤ ..... ) FAIRE
      B[k]=A[i]
      i ← i+1
      k ← k+1
    FIN TANT QUE
    On a atteint la fin du premier tableau, mais il reste des éléments du
deuxième tableau : on complète le tableau B avec ces éléments
    TANT QUE (j ≤ ..... ) FAIRE
      B[k]=A[j]
      j ← j+1
      k ← k+1
    FIN TANT QUE
    A ← B
    retourner(A)
  FIN
```

2.3 Le tri fusion :exemple(1)

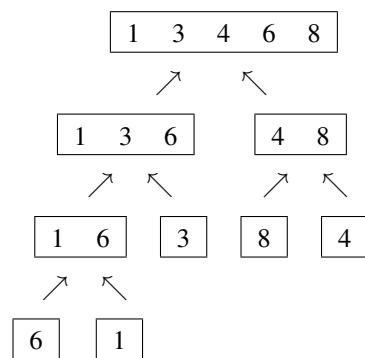
Trier les éléments du tableau :

6	1	3	8	4
---	---	---	---	---

Les divisions successives ...



Les fusions successives ... en commençant par le bas



Un autre exemple avec animation : http://lwh.free.fr/pages/algo/tri/tri_fusion.html