



IEEA

Dans toute cette feuille on suppose que le type `TABLEAU` a été déclaré à l'aide des déclarations suivantes :

```
type INDICE = ..... ; // un intervalle d'entiers
ELEMENT = ..... ; // un type auquel on peut appliquer la relation d'ordre <=
TABLEAU = array[INDICE] of ELEMENT ;
```

Exercice 1 : *Dérouler l'algorithme du tri fusion*

Étant donné le tableau `t`

4	3	7	1	2	5	3	2
---	---	---	---	---	---	---	---

donnez les valeurs successives que prend ce tableau `t` lors du tri fusion.

Exercice 2 : *Implémenter le tri fusion en Pascal*

Q 1 . Écrivez la procédure `fusionner` à quatre paramètres `t`, le tableau, et `a`, `b`, `c`, trois indices, qui fusionne les deux tranches `t[a..b]` et `t[b+1..c]` qui sont supposées triées.

Q 2 . Écrivez la procédure `trier_fusion_rec` à trois paramètres `t`, le tableau, et `a`, `b`, deux indices qui trie la tranche de tableau `t[a..b]` en utilisant l'algorithme du tri fusion.

Q 3 . Comment s'écrit alors la procédure `trier_fusion` à un paramètre `t`, le tableau à trier ?

Exercice 3 : *Pire et meilleur des cas*

Q 1 . Trouvez deux tableaux $t_{meilleur}$ et t_{pire} de taille 8, qui réalisent respectivement le meilleur et le pire des cas pour le tri fusion.

Exercice 4 : *Un autre algorithme de tri*

Voici un algorithme de tri très efficace puisqu'il est linéaire, malheureusement il n'est pas toujours utilisable car il faut que la clef du tri soit d'un type discret et ne puisse prendre que peu de valeurs, par exemple on trie des copies suivant les notes qui sont entières et comprises entre 0 et 10.

Une idée très simple est de faire un parcours des copies en faisant un tas pour chaque note, donc un tas avec les 0, un autre avec les 1, un troisième tas avec les 2 et ainsi de suite ; à la fin de ce partage on a 11 tas qu'il suffit de regrouper dans l'ordre qu'on désire, croissant ou décroissant.

Le tri *comptage* (ou par *casiers*) s'inspire de cette technique.

Étant donné un tableau `t` à trier, on commence par parcourir ce tableau en comptant combien il y a d'éléments pour chaque valeur de la clef, on fait cela en consignait les résultats dans un tableau d'effectifs qui est indicé par les différentes valeurs de la clef. Ensuite on transforme ce tableau d'effectifs en tableau d'effectifs cumulés. Enfin grâce à ces effectifs cumulés on peut recopier le tableau sous sa forme trié.

Exemple :

- On suppose qu'on a 1000 copies avec des notes entières entre 0 et 10
- Après comptage on a le tableau des effectifs suivant :

effectifs :	16	27	74	135	196	200	175	96	43	25	13
notes :	0	1	2	3	4	5	6	7	8	9	10

- Ce qui nous donne pour les effectifs cumulés

effectifs :	16	43	117	252	448	648	823	919	962	987	1000
notes :	0	1	2	3	4	5	6	7	8	9	10

- On sait alors que les 16 premières cases seront occupées par les copies ayant 0, puis les cases d'indices 17 à 43 le seront par des copies ayant 1 et ainsi de suite.

Q 1 . Écrivez une procédure de tri qui utilise cet algorithme.