

Auteur : Jean-Marc Duquesnoy et Raymond Moché

1 - Apparemment, on peut ranger deux nombres donnés a et b à l'aide de l'algorithme suivant :

Algorithme 1 : Écrire 2 nombres a et b dans l'ordre croissant

Entrées :

nombres a et b ;

Sorties :

minimum puis maximum des nombres a et b ;

début

```

si  $a > b$  alors
  |  $a \leftarrow b$  ;
  |  $b \leftarrow a$  ;
fin
  Afficher  $a, b$ 
fin
    
```

$a \leftarrow b$ signifie que le contenu de la case b , qui est le nombre b , vient se mettre à la place du contenu de la case a (qui était le nombre a). En sortie, on demande d'afficher successivement les contenus des cases a et b .

Question - Pourquoi cet algorithme est-il incorrect ? Quel résultat donne-t-il ?

2 - Voici un algorithme correct pour ordonner a et b dans l'ordre croissant :

Algorithme 2 : Écrire 2 nombres a et b dans l'ordre croissant

Entrées :

nombres a et b ;

Sorties :

minimum puis maximum des nombres a et b ;

début

```

si  $a > b$  alors
  | échanger  $a$  et  $b$  ;
fin
  Afficher  $a, b$ 
fin
    
```

On voit que le problème est d'échanger le contenu des 2 cellules a et b .

Question - Donner une solution utilisant une cellule auxiliaire c . La réponse sera donnée en complétant le cadre vide de l'algorithme ci-dessous :

Algorithme 3 : Écrire 2 nombres a et b dans l'ordre croissant

Entrées :

nombres a et b ;

Sorties :

minimum puis maximum des nombres a et b ;

début

```

si  $a > b$  alors
  | 
fin
  Afficher  $a, b$ 
fin
    
```

3 - Qu'est-ce que le « tri à bulles » ?

Supposons que l'on ait à ordonner dans l'ordre croissant la suite

$$A = 5, 1, 4, 8, 2.$$

La *manipulation de base* du « tri à bulles » consiste à examiner *successivement* les couples formés par les nombres de rang 1 et 2, puis par les nombres de rang 2 et 3 et ainsi de suite jusqu'à la fin et de leur appliquer le traitement suivant :

On laisse inchangé tout couple qui est déjà rangé dans l'ordre croissant ; dans le cas contraire, on échange ses 2 termes.

Voici les transformations successives subies par A au cours de ce processus :

$$A = \underline{5}, 1, 4, 8, 2 \rightarrow 1, \underline{5}, 4, 8, 2 \rightarrow 1, 4, \underline{5}, 8, 2 \rightarrow 1, 4, 5, \underline{8}, 2 \rightarrow B = 1, 4, 5, 2, 8$$

La suite B n'est pas ordonnée dans l'ordre croissant mais *nous avons fait des progrès* dans le classement car le maximum de $5, 1, 4, 8, 2$, qui est 8, se trouve maintenant à la fin de la suite, qui est sa place définitive.

Cela conduit naturellement à appliquer à B la *manipulation de base*. Voici ce que cela donne

$$B = \underline{1}, 4, 5, 2, 8 \rightarrow 1, \underline{4}, 5, 2, 8 \rightarrow 1, 4, \underline{5}, 2, 8 \rightarrow 1, 4, 2, \underline{5}, 8 \rightarrow C = 1, 4, 2, 5, 8$$

C'est mieux car les 2 derniers termes sont maintenant à leurs places définitives. Re commençons en appliquant la manipulation de base à C :

$$C = \underline{1}, 4, 2, 5, 8 \rightarrow 1, \underline{4}, 2, 5, 8 \rightarrow 1, 2, \underline{4}, 5, 8 \rightarrow 1, 2, 4, \underline{5}, 8 \rightarrow 1, 2, 4, 5, 8$$

On voit que le rangement de A dans l'ordre croissant est terminé. Pour cela, on a dû appliquer 3 fois la manipulation de base.

Si on imagine maintenant que A est une suite quelconque de $n \geq 2$ nombres quelconques, il est clair qu'après une première application de la *manipulation de base*, le dernier terme du nouvel état de A sera à sa place définitive ; après la seconde application de la *manipulation de base*, les 2 derniers termes seront à leurs places définitives, et ainsi de suite. Par conséquent, après la $(n - 1)^{\text{ème}}$ application de la *manipulation de base*, les $(n - 1)$ derniers termes - et par conséquent le premier - seront à leurs places définitives, ce qui veut dire que le rangement sera terminé. L'exemple $A = 5, 1, 4, 8, 2$ montre que, quelquefois, il n'y a pas besoin d'effectuer les $(n - 1)$ applications de la *manipulation de base*.

Question - Combien faut-il de *manipulations de base* pour ranger $5, 4, 3, 2, 1$ dans l'ordre croissant ?

Nous retiendrons que nous sommes sûrs que A est rangé dans l'ordre croissant au bout de $(n - 1)$ *manipulations de base*.

4 - Voici l'algorithme du « tri à bulles », en langage naturel :

Algorithme 4 : Algorithme du tri à bulles

Entrées :

$A = a_1, \dots, a_n$, suite d'au moins 2 nombres ;

Sorties :

la suite obtenue en rangeant A dans l'ordre croissant ;

début

```
 $n \leftarrow$  longueur de la suite  $A$  ;  
pour  $k = 1, \dots, (n - 1)$  faire  
  pour  $j = 1, \dots, (n - 1)$  faire  
    si  $a_j > a_{j+1}$  alors  
       $c \leftarrow a_j$  ;  
       $a_j \leftarrow a_{j+1}$  ;  
       $a_{j+1} \leftarrow c$   
    fin  
  fin  
fin  
Afficher  $A$   
fin
```

Question - Modifier cet algorithme pour que l'état initial de la suite A soit conservé et affiché en même temps que le résultat.

5 - Ouvrir le fichier « TriBulles.sci » qui contient l'algorithme du « tri à bulles » sous une forme exécutable par « scilab pour les lycées » ou le fichier « TriBullesXcas.xws » exécutable par « Xcas ».

5.a - À l'aide de cet algorithme, ranger $A = 5, 1, 4, 8, 2$ dans l'ordre croissant.

5.b - Tirer 300 entiers au hasard entre 213 et 945 ; les ranger dans l'ordre croissant.

On remarquera que ce calcul est long.

5.c - Recommencer le tri en utilisant la fonction pré-programmée « trier » (« sort ») de « scilab pour les lycées » ou « SortD » de « Xcas ». Que constate-t-on ?

Remarque Les fonctions « trier » et « SortD » trient dans l'ordre décroissant.

5.d - Tirer au hasard 100000 nombres quelconques entre 0 et 1. Ranger ces nombres dans l'ordre croissant.

Commentaires Ce tri aurait été impossible (essayer) avec l'algorithme du « tri à bulles » tel que nous l'avons écrit. L'algorithme du « tri à bulles », même optimisé, n'est pas très bon. Les spécialistes font beaucoup mieux.