

Méthodes de tri - corrigé

Pour chacune des méthodes de tri, proposer un algorithme et coder cet algorithme en langage Maple. On choisira arbitrairement un tableau d'une dizaine de nombres entiers que l'on devra trouver trié par ordre croissant après exécution du programme.

1 - Tri à bulle

Le tri à bulle consiste à faire remonter progressivement les éléments les plus légers. En parcourant le tableau on compare chaque élément à son suivant et on les permute s'il y a lieu.

Après une première passe, l'élément le plus lourd est assurément au bas du tableau. A la passe suivante il est alors possible de ne parcourir le tableau que jusqu'à l'avant dernier élément, et ainsi de suite jusqu'à la dernière passe qui consistera à comparer le premier élément au deuxième...

```
> t:=[5,4,6,10,1,3];
  for i_max from 5 to 1 by -1
    do for i from 1 to i_max
      do if t[i]>t[i+1]
        then memoire:=t[i]:
          t[i]:=t[i+1]:
          t[i+1]:=memoire
        fi
      od
    od;
  t;
```

t := [5, 4, 6, 10, 1, 3]
[1, 3, 4, 5, 6, 10]

2 – Tri par sélection

Le tri par sélection consiste à sélectionner l'élément le plus petit parmi ceux qui restent à trier et à le placer immédiatement à sa place définitive. En parcourant le tableau, on recherche tout d'abord l'élément le plus petit et on le met directement à sa place en réalisant l'échange avec le premier élément du tableau. On recommence de même à partir du deuxième élément et ainsi de suite jusqu'à identifier le plus petit des deux derniers éléments du tableau...

```
> t:=[5,4,6,10,1,3];
  for rang from 1 to 5
    do mini:=rang:
      for i from rang+1 to 6
        do if t[i]<t[mini] then mini:=i
          fi
        od:
      memoire:=t[rang]:
      t[rang]:=t[mini]:
      t[mini]:=memoire
    od;
  t;
```

t := [5, 4, 6, 10, 1, 3]
[1, 3, 4, 5, 6, 10]

3 - Tri par insertion

Le tri par insertion consiste à prendre les éléments un par un en commençant par la fin et à les placer où il convient dans le début de liste triée en décalant vers la fin autant que nécessaire les éléments déjà triés.

```
> t:=[5,4,6,10,1,3];
   for i from 2 to 6
     do element:=t[i]:
       j:=i-1:
       while j>0 and t[j]>element do t[j+1]:=t[j]:
                                     j:=j-1
                                   od:
       t[j+1]:=element
     od:
   t;
```

$t := [5, 4, 6, 10, 1, 3]$

$[1, 3, 4, 5, 6, 10]$