

## Objectif :

Découvrir une autre méthode de tri : Tri par fusion

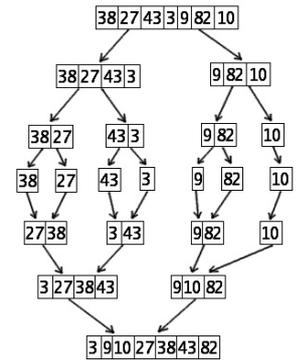
Les différents types de tris sont testables sur le site :

<http://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

## 1. Tri par fusion:

C'est une combinaison des tris précédents. Son exécution est la suivante :

- On coupe en deux parties à peu près égales les données à trier.
  - On trie les données de chaque partie (pour cela, on coupe chaque partie en deux et on trie chacune)
  - On fusionne les deux parties
- La récursivité s'arrête car on finit par arriver à des listes composées d'un seul élément et le tri est alors trivial.



### a) Algorithme :

Ci-dessous l'algorithme du tri par fusion :



## b) Test :

I) Tester "à la main", cet algorithme pour le tableau suivant {7,3,1,6,2,5} en complétant le tableau ci-dessous :

i	0	1	2	3	4	5
Tableau	731625	371625				

## c) Programmation :

II) Ecrire le programme correspondant à cet algorithme avec le même tableau qu'en I et enregistrer sous II"votre nom".pde

On donne la fonction fusion :

```
void fusion(int tableau[],int deb1,int fin1,int fin2)
{
    int deb2=fin1+1;

    //on recopie les éléments du début du tableau
    int table1[]=new int[fin1-deb1+1];
    for(int i=deb1;i<=fin1;i++)
    {
        table1[i-deb1]=tableau[i];
    }
    int compt1=deb1;
    int compt2=deb2;
    for(int i=deb1;i<=fin2;i++)
    {
        if (compt1==deb2) //c'est que tous les éléments du premier tableau ont été utilisés
        {
            break; //tous les éléments ont donc été classés
        }
        else if (compt2==(fin2+1)) //c'est que tous les éléments du second tableau ont été utilisés
        {
            tableau[i]=table1[compt1-deb1]; //on ajoute les éléments restants du premier tableau
            compt1++;
        }
        else if (table1[compt1-deb1]<tableau[compt2])
        {
            tableau[i]=table1[compt1-deb1]; //on ajoute un élément du premier tableau
            compt1++;
        }
        else
        {
            tableau[i]=tableau[compt2]; //on ajoute un élément du second tableau
            compt2++;
        }
    }
}
```

## 2. Comparaison avec les tris de la séance 10

III) Tester votre programme avec le tableau de 1000 éléments créé par le programme :

```
int[] a = new int [10000];
for(int k=0; k < a.length; k=k+1)
{
    a [k] = int (random(1000));
}
```

Ensuite pour comparer, il faut modifier pour déterminer le temps d'exécution à l'aide de la fonction millis() qui retourne le temps écoulé depuis le début du programme.

Ajouter la fonction millis().

N'afficher qu'une seule fois le tableau à la fin.

Enregistrer vos programme sous IIItrifusion.pde.

Lequel est le plus efficace (vous pouvez vous aider <http://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>).