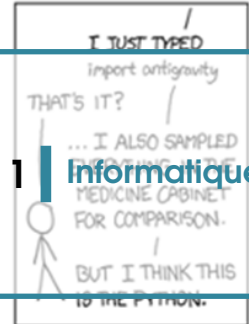
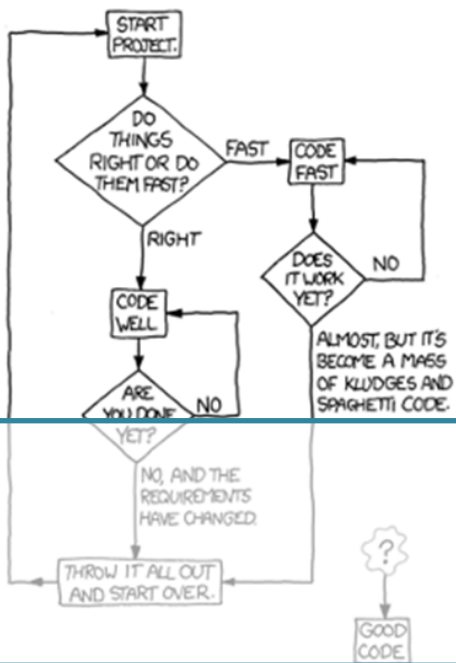


## HOW TO WRITE GOOD CODE:



Semestre 1 | Informatique

## Thèmes d'étude

1	Présentation	2
2	Tris	2
3	Activité préparatoire	3
4	QCM	4
5	TP	5

Thème : Tris.

Commentaires :

- algorithmique quadratique : tri par insertion, par sélection;
- tri par partition-fusion;
- tri par comptage.

*On fait observer différentes caractéristiques (par exemple stable ou non, en place ou non, comparatif ou non ...).*

## 1 Présentation

Le tri de données ou de valeurs est omniprésent en informatique. Pour cela, beaucoup d'algorithmes ont été développés afin de réaliser des tris rapidement, notamment lorsque le nombre de données est important.

### Définition Stabilité

#### Définition Tri en place

Un tri est effectué en place lorsque la liste à trier est modifiée jusqu'à devenir triée. Dans le cas contraire, la fonction de tri pourra renvoyer une nouvelle liste contenant les mêmes éléments, mais triés.

#### Définition Tri comparatif

Un tri est dit comparatif lorsqu'il s'appuie uniquement sur la comparaison deux à deux des éléments de la liste et pas sur la valeur de ces éléments.

## 2 Tris

### Définition Tri par insertion

À partir d'une sous-liste triée, le tri par insertion consiste à parcourir les éléments non triés et de les insérer successivement dans la sous-liste déjà triée.

```
def insere(t, j):
    k, a = j, t[j]
    while k > 0 and a < t[k-1]:
        t[k] = t[k-1]
        k = k-1
    t[k] = a

def insertionSort(t):
    for j in range(1, len(t)):
        insere(t, j)
```

### Définition Tri rapide

Soit une liste  $L$  non triée. Soit  $p$  un terme appelé pivot. Le tri rapide consiste à répartir les éléments strictement inférieur au pivot avant ce dernier et les termes plus grand après le pivot (segmentation). Le pivot est à ce stade trié correctement par rapport aux autres valeurs de la liste. Ce principe est alors appliqué récursivement aux deux sous-listes séparées par le pivot.

```
def segmente(t, i, j):
    p = t[j-1] # On prend comme pivot le dernier élément de la sous liste.
    a = i
    for b in range(i, j-1):
        if t[b] < p:
            t[a], t[b] = t[b], t[a]
            a += 1
    t[a], t[j-1] = t[j-1], t[a] # On positionne le pivot "à sa place".
    return a # On retourne l'index du pivot. Le tableau a été modifié en place.
```

```
def quickSort(t, i, j):
    if i + 1 < j:
        a = segmente(t, i, j)
        quickSort(t, i, a)
        quickSort(t, a + 1, j)
```

```
# Instruction pour trier une liste
quickSort(t, 0, len(t))
```

### Définition Tri fusion

Il s'agit d'un tri s'appuyant sur la stratégie diviser pour régner.

L'algorithme est le suivant :

- on divise la liste en deux listes de tailles quasi-identiques;
- on trie récursivement ces deux listes;
- on fusionne les deux listes triées.

```
def placer(L :list, p :int, x) :
    """Place un élément x à sa place dans une liste L triée à partir de l'indice p
    Entrée :
        L : une liste, p : un entier, x : un élément
    Sorties :
        La liste est modifiée mais n'est pas renvoyée. k la valeur de l'indice de la liste où l'
        élément a été placé"""
    k = p
    while ( k < len(L) and x > L[k]) :
        k = k+1
    L.insert(k, x)
    return k

def fusion(a:list, b:list) :
    """Fusionne les deux listes
    Entrée : deux listes a et b triées.
    Sortie : La liste b modifiée"""
    p = 0
    for x in a :
        p = placer(b, p, x)+1

def tri_fusion(t : list) :
    """Trie la liste t
    Entrée : une liste.
    Sortie : la liste est modifiée."""
    if len(t) < 2 :
        return (t)
    else :
        m = len(t) // 2
        return (fusion (tri_fusion(t[:m]) , tri_fusion(t[m:]) ))
```

**Proposition Complexité des algorithmes** On note  $T(n)$  le nombre de comparaisons nécessaire pour trier une liste de longueur  $n$ . On montre que dans le pire des cas, les complexités sont les suivantes :

- tri par insertion :  $T_{\text{Max}}(n) = \mathcal{O}(n^2)$ ;
- tri rapide :  $T_{\text{Max}}(n) = \mathcal{O}(n^2)$ ;
- tri partition-fusion :  $T_{\text{Max}}(n) = \mathcal{O}(n \log n)$ .

## 3 Activité préparatoire

Pour réaliser l'activité associée à ce cours, suivre le lien suivant :

- Sujet :
- Corrige :



## 5 TP