

A retenir du chapitre 15 :

Algorithmes de tri 2



- Algorithme de tri à bulles :

```
def TriBulles(l) :
    n=len(l)
    for i in range(n-1) :
        for j in range(0,n-i-1) :
            if l[j]>l[j+1] :
                l[j],l[j+1]=l[j+1],l[j]
    return l
```

Sa complexité est quadratique.

- Algorithme de tri par sélection :

```
def TriSelection(l) :
    n=len(l)
    for i in range(n) :
        imin=i #recherche du minimum à partir du rang i
        m=l[i]
        for j in range(i+1,n) :
            if l[j]<m :
                imin=j
                m=l[j]
        l[imin],l[i]=l[i],l[imin] # on place le minimum en position i
    return l
```

Sa complexité est quadratique.

- Algorithme de tri par insertion :

```
def TriInsertion(l) :
    n=len(l)
    for i in range(n) :
        k=i
        cle=l[i]
        while k>0 and cle<l[k-1] :
            l[k]=l[k-1] #décalage vers la droite des valeurs plus grandes que la clé
            k-=1
        l[k]=cle# insertion de la clé
    return l
```

Sa complexité est quadratique.

- On peut améliorer la complexité des algorithmes de tri en utilisant la stratégie diviser pour régner comme dans le tri fusion qui a une complexité $O(n \ln n)$.