

Algorithmes de tri

Fiche de synthèse

① GÉNÉRALITÉS

Un **algorithme de tri** range les éléments d'une liste du plus petit au plus grand, en les comparant deux à deux. Entiers et flottants se comparent par $\leq$; les chaînes de caractères par l'ordre lexicographique, où toutes les majuscules passent avant les minuscules : 'Z' < 'a' vaut `True`.

Définition 1 (Tri en place, tri stable)

- Un tri est **en place** s'il travaille dans la liste elle-même, sans en créer une autre.
- Un tri est **stable** s'il garde dans leur ordre de départ les éléments égaux pour le critère de tri.

Des élèves rangés par ordre alphabétique, triés par note avec un tri stable, restent dans l'ordre alphabétique à note égale : on peut trier selon un critère, puis un autre, sans défaire le premier classement.

```
def echanger(L, i, j):
    L[i], L[j] = L[j], L[i]
    return L

def est_triee(L):
    for i in range(len(L) - 1):
        if L[i] > L[i + 1]:
            return False
    return True
```

② TRI PAR SÉLECTION

On cherche le plus petit élément et on l'échange avec le premier; puis on cherche le plus petit du reste et on l'échange avec le deuxième, et ainsi de suite jusqu'à l'avant-dernier.

départ	12	9	3	7	14	11
$i = 0$	3	9	12	7	14	11
$i = 1$	3	7	12	9	14	11
$i = 2$	3	7	9	12	14	11
$i = 3$	3	7	9	11	14	12
$i = 4$	3	7	9	11	12	14

```
def tri_selection(L):
    for i in range(len(L) - 1):
        indice_du_min = i
        for j in range(i + 1, len(L)):
            if L[j] < L[indice_du_min]:
                indice_du_min = j
        echanger(L, i, indice_du_min)
    return L
```

Invariant : après le tour i , les $i + 1$ premiers éléments sont triés et à leur place définitive. Le tri par sélection est en place, mais pas stable.

③ TRI PAR INSERTION

Comme on range des cartes à jouer : le début de la liste est trié, et l'on fait remonter l'élément suivant jusqu'à sa place parmi ceux qui le précèdent.

```
def tri_insertion(L):
    for i in range(1, len(L)):
        en_cours = L[i]
        j = i
        while j > 0 and L[j - 1] > en_cours:
            L[j] = L[j - 1]
            j -= 1
        L[j] = en_cours
    return L
```

Pour [12, 9, 3, 7, 14, 11], la partie triée devient [9, 12], puis [3, 9, 12], [3, 7, 9, 12], [3, 7, 9, 12, 14] et enfin [3, 7, 9, 11, 12, 14]. Invariant : après le tour i , les $i + 1$ premiers éléments sont triés, pas forcément à leur place définitive. Le tri par insertion est en place et stable.

④ COMPLEXITÉ

Le tri par sélection compare toujours $(n-1) + (n-2) + \dots + 1 = \frac{n(n-1)}{2}$ fois. Le tri par insertion ne fait que $n-1$ comparaisons sur une liste déjà triée, mais $\frac{n(n-1)}{2}$ décalages sur une liste triée à l'envers.

Tri	Meilleur	Moyen	Pire	Stable	En place
sélection	$O(n^2)$	$O(n^2)$	$O(n^2)$	non	oui
insertion	$O(n)$	$O(n^2)$	$O(n^2)$	oui	oui

À retenir.

- Sélection : placer à chaque tour le plus petit des éléments restants.
- Insertion : insérer chaque élément à sa place dans le début déjà trié.
- Les deux sont quadratiques; l'insertion est linéaire sur une liste déjà triée.