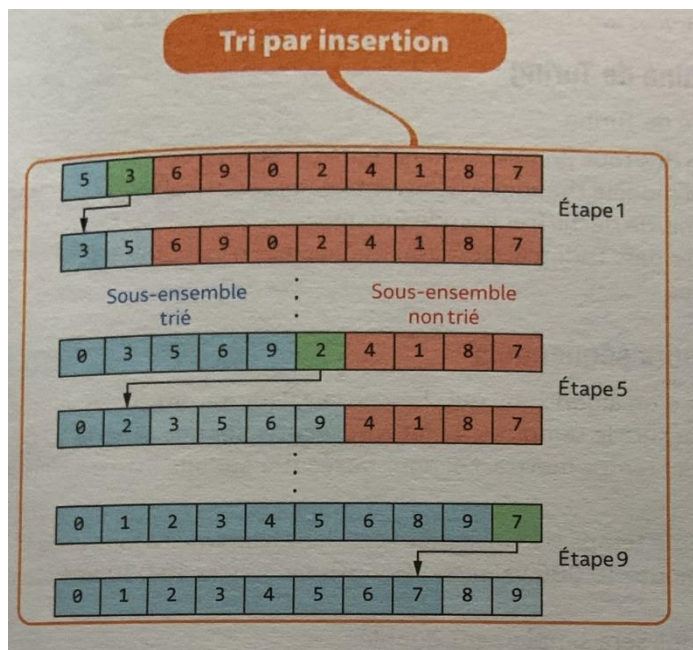


Tri par insertion

Principe

Le tri par insertion considère chaque élément du tableau et l'insère à la bonne place parmi les éléments déjà triés. Ainsi, au moment où on considère un élément, les éléments qui le précèdent sont déjà triés, tandis que les éléments qui le suivent ne sont pas encore triés.



Correction

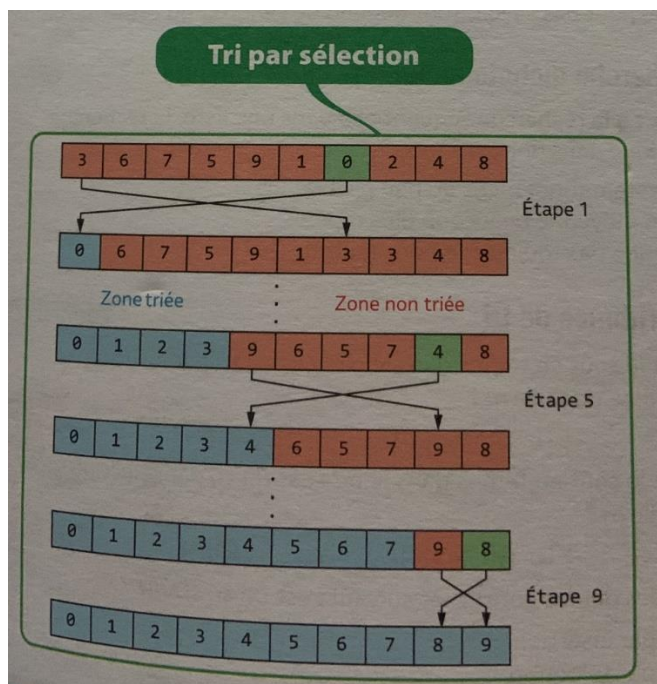
```
def tri_insertion(lst:list) :  
    for j in range(1, len(lst)):  
        cle = lst[j]  
        i = j-1  
        while i >=0 and lst[i] > cle :  
            lst[i+1] = lst[i]  
            i = i-1  
        lst[i+1] = cle  
  
a = [9,8,7,6,5,4,3,2,1]  
tri_insertion(a)  
assert a == [1,2,3,4,5,6,7,8,9]
```

Tri par sélection

Principe

Le principe du tri par sélection est le suivant :

- Rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 0 ;
- Rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1 ;
- Continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.



Correction

```
def tri_selection(lst:list):  
    for i in range(0, len(lst)-1):  
        # recherche le + petit element de i à la fin  
        mini = i  
        for j in range(i+1, len(lst)):  
            if lst[j] < lst[mini] :  
                mini = j  
        # echanger les cases i et mini  
        lst[i],lst[mini] = lst[mini], lst[i]  
  
a = [9,8,7,6,5,4,3,2,1]  
tri_selection(a)  
assert a == [1,2,3,4,5,6,7,8,9]
```