

Recherche séquentielle, recherche linéaire:

```
def recherche1(T, x)->int:
    """
    :param T: list of elements
    :param x: element
    :return: int, index of x in the list
             else -1
    """
    i = 0
    while i < len(T) and T[i] != x:
        i = i + 1
    if i == len(T):
        return -1
    else:
        return i
```

Recherche dichotomique:

```
def recherche_dicho(T,x):
    """
    :param T: sorted list of elements
    :param x: element
    :return: int, index of x in the list else -1
    """
    i_min = 0
    i_max = len(T) - 1
    while (i_max >= i_min):
        mid = (i_min + i_max)//2
        if T[mid] == x:
            return mid
        elif T[mid] < x:
            i_min = mid + 1
        else:
            i_max = mid - 1
    return -1
```

Recherche du minimum:

il existe une fonction native en python qui retourne la valeur min dans une liste:

```
>>> min([0,-10, 5, 3])
-10
```

mais celle-ci ne retourne pas le RANG du minimum. Voici le programme de la fonction recherche_du_min qui retourne l'indice du min ainsi que la valeur min d'une liste:

```
def recherche_du_min(L):
    mini = L[0]
    ind_mini = 0
    for i in range(len(L)):
        if L[i] < mini:
            mini = L[i]
            ind_mini = i
    return ind_mini, mini
```

```
>>> recherche_du_min([0,-10, 5, 3])
1, -10
```

Recherche du maximum:

il existe une fonction native en python qui retourne la valeur max dans une liste:

```
>>> max([0,-10, 5, 3])  
5
```

mais celle-ci ne retourne pas le RANG du maximum. Voici le programme de la fonction recherche_du_max qui retourne l'indice du max ainsi que la valeur max d'une liste:

```
def recherche_du_max(L):  
    maxi = L[0]  
    ind_maxi = 0  
    for i in range(len(L)):  
        if L[i] > maxi:  
            maxi = L[i]  
            ind_maxi = i  
    return ind_maxi, maxi
```

```
>>> recherche_du_max([0,-10, 5, 3])  
2, 5
```

Tri par insertion:

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

```
>>> tri_insertion([49, 4, 89, 13, 44, 60, 12, 2, 5, 62])  
[2, 4, 5, 12, 13, 44, 49, 60, 62, 89]
```

Tri par selection

```
def imin(T,a,b):  
    """fonction qui cherche la valeur  
    minimum entre les positions a et b  
    """  
    imin = a  
    N = len(T)  
    for i in range(a+1,b):  
        if T[i] < T[imin]:  
            imin = i  
    return imin
```

```
def triselection(T):  
    N = len(T)  
    for i in range(N-1):  
        j = imin(T,i,N)  
        T[i],T[j] = T[j],T[i]
```