INFO-H-100 - Programmation
TP 11 — Tris

Lorsque l'exercice demande d'écrire une fonction, écrivez la fonction demandée et testez-la avec plusieurs valeurs pertinentes.

Ex. 1. Écrire une fonction `selection_sort` qui trie de manière croissante une liste donnée en paramètre avec l'algorithme du tri par sélection. Par exemple :

```
ls = [2, 5, 9, 9, 6, 6, 4, 5, 2, 3, 1, 5, 8]
selection_sort(ls)
print(ls)           # -> [1, 2, 2, 3, 4, 5, 5, 5, 6, 6, 8, 9, 9]
```

Ex. 2. Écrire une fonction `insertion_sort` qui trie de manière croissante une liste donnée en paramètre avec l'algorithme du tri par insertion.

Ex. 3. Écrire une fonction `in_list` qui détermine si, oui ou non, une valeur donnée existe dans une liste triée. Utiliser la recherche dichotomique.

Ex. 4. Écrire une fonction `poly_sort` qui reçoit une liste de polynômes, et une valeur x et qui trie cette liste en se basant sur l'ordre défini par la valeur des polynômes en x. Par exemple :

```
p1 = [4, 2, 3]
p2 = [2, 1]
p3 = [2, -3, 4, 1]
p4 = [-1, 2]
ls = [p1, p2, p3, p4]
poly_sort(ls, 2)
print(ls) # -> [[-1, 2], [2, 1], [2, -3, 4, 1], [4, 2, 3]]
```

Vous utiliserez le tri par sélection.

Les polynômes sont représentés comme vu au cours et lors de la séance de TP 9. Reprenez la fonction `p_eval` de l'exercice 8 de cette séance.

INFO-H-100 - Programmation
TP 11 — Tris
Corrections

Solution de l'exercice 1:

```
def min_pos_from(ls, i):
    """La position du min de ls en commencant en i.
    Pre: i dans ls
    """
    res = i
    while i < len(ls):
        if ls[i] < ls[res]:
            res = i
        i += 1
    return res

def swap(ls, i1, i2):
    ls[i1], ls[i2] = ls[i2], ls[i1]

def selection_sort(ls):
    for i in range(len(ls) - 1):
        pos = min_pos_from(ls, i)
        swap(ls, i, pos)
```

Solution de l'exercice 2:

```
def move(ls,i,j):
    """Deplace, dans ls, la valeur en position i et l'insert en position j
    Pre: i >= j
    """
    val = ls[i]
    del ls[i]
    ls.insert(j,val)

def insertion_pos(val, ls, n):
    """La position d'insertion de val parmi les n 1er elements de ls
    n si val est le plus grand
    """
    j = 0
    while j < n and val > ls[j]:
        j += 1
    return j

def insertion_sort(ls):
    for i in range(1, len(ls)):
        j = insertion_pos(ls[i], ls, i)
        move(ls, i, j)
```

ou

```
def insertion_sort(ls):
    for i in range(1, len(ls)):
        val = ls[i]
        j = i
        while j > 0 and ls[j - 1] > val:
            ls[j] = ls[j - 1]
            j -= 1
        ls[j] = val
```

Solution de l'exercice 3:

```
def in_list(ls, val):
    """True ssi val apparait dans ls
    Pre: ls est triee
    """
    bi = 0
    bs = len(ls) - 1
    m = (bs + bi) / 2
    while bi <= bs and ls[m] != val:
```

```
        if val < ls[m]:
            bs = m - 1
        else:
            bi = m + 1
        m = (bs + bi) / 2
    return bi <= bs
```

Solution de l'exercice 4:

```
def swap(ls, i1, i2):
    ls[i1], ls[i2] = ls[i2], ls[i1]

def min_pos_from(ls, i, x):
    """La position du min de ls en commencant en i.
    Pre: i dans ls
    """
    res = i
    while i < len(ls):
        if p_eval(ls[i], x) < p_eval(ls[res], x):
            res = i
        i += 1
    return res

def poly_sort(ls, x):
    for i in range(len(ls) - 1):
        pos = min_pos_from(ls, i, x)
        swap(ls, i, pos)

def p_eval(poly,x):
    """ Eval le polynome poly en x """
    val = 0
    facteur = 1
    for coeff in poly:
        val += coeff * facteur
        facteur *= x
    return val
```