

|  |      |           |     |
|--|------|-----------|-----|
| NSIT   | e07a | Révisions |     |
| Kata 1   |      |           |     |
| Tests unitaires, spécifications, listes, récursivité |      |           | 1/1 |

**Exercice :** Tri par sélection dans l'ordre décroissant

**Énoncé :** C'est écrit dans le titre ;)

```
def tri_selection(ma_liste:list)->list:
    """ Renvoie une copie triée dans m'ordre croissant de ma_liste
        Précondition: ma_liste n'est pas vide
        Post_condition: liste_triee a le même nombre d'éléments que ma_liste
    """
    # On initialise la liste_triee vide
    liste_triee = []
    # On copie la liste d'origine pour ne pas la détériorer
    ma_liste_copie = ma_liste.copy() # Evite les effets de bord
    # Tant qu'il reste un élément dans ma_liste_copie
    while len(ma_liste_copie)>0:
        # On trouve le maximum et sa position dans ma_liste grâce à maximum_position() ou maximum()
        ...
        # Ajoute ce maximum à liste_triee
        ...
        # Retire ce maximum à ma_liste_copie
        ...
    return liste_triee

assert tri_selection([3,5,1,2]) == [5,3,2,1], 'Le test n\'est pas passé'
L = [randint(1,100) for i in range(1,20)] # Tire une liste aléatoire
print(L)
tri_selection(L)
```

**Commentaire :**

Pour voir une démonstration du fonctionnement des algorithmes de tris, rendez-vous sur le site <http://visualgo.net> dans la rubrique 'Sorting'. Dans cette animation par contre, le tri se fait dans l'ordre croissant.

`liste_triee` est initialisée par la liste vide. Elle sera alimentée au fur et à mesure des plus grandes valeurs restantes trouvées sur la liste de travail.

Attention, on ne travail pas directement sur la liste passée en paramètre (`ma_liste`), mais sur une copie de cette liste :

`ma_liste_copie = ma_liste.copy()`

Puisque nous allons au fur et à mesure supprimer le plus grand élément de la liste pour le rajouter à `liste_triee`, nous ne voulons pas créer un effet de bord et modifier la liste d'origine.

**Remarque :** En termes de complexité (en temps), le tri par sélection est un très mauvais tri.

Mais il fait le travail et il est facile à programmer.