

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

**Exercice** : Interdépendance des fonctions

**Énoncé** : De quels définitions dépend le bon fonctionnement d'une autre fonction

**Commentaire** :

Dans le cas de l'exemple, l'appel de la fonction `maximum_rec()` fait bien appel à `sans_tete()`

```
def maximum_rec(ma_liste:list, m:float)->float:
    """ Renvoie le maximum de la liste ma_liste
        Précondition: ma_liste n'est pas vide.
    """
    # Cas de base: Si la taille de ma_liste = 1, alors, renvoie le maximum entre l'élément restant et m
    if len(ma_liste)==1:
        return max(ma_liste[0], m)
    # Sinon, renvoie maximum_rec(ma_liste privée de la tête, maximum entre la tête de la liste et m)
    else:
        nouveau_max = max(ma_liste[0], m)
        ma_liste_sans_tete = sans_tete(ma_liste) # Copie la liste privée du premier élément.
        return maximum_rec(ma_liste_sans_tete, nouveau_max)

L = [randint(1,100) for i in range(1,20)] # Tire une liste aléatoire
print(L)
assert maximum_rec([1,2,5,3],1)==5, 'La fonction maximum ne renvoie pas le bon résultat sur une \
liste d\'entiers'
assert maximum_rec([1.8,2.4,5.1,3.1],1.8)==5.1, 'La fonction maximum ne renvoie pas le bon résultat \
sur une liste de flottants'
maximum_rec(L,L[0])
```

Dans le code de la fonction `extremum()`, on retrouve l'appel à la fonction `tri_selection()` qui elle même fait appel à `maximum_position()`