

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

Exercice : Compare les performances de deux tris

Énoncé : Utilisation de la bibliothèque `time` pour chronométrer le temps d'exécution des appels de fonctions

```
def compare_les_tris(N:int, taille:int)->tuple:
    """ Crée une liste de 'taille' éléments
        Chronomètre l'exécution de N tris du tri sélection et du tri par sorted()
        Renvoie le couple constitué des deux temps
    """
    # Importe la bibliothèque time
    import time
    # Crée une liste aléatoire de 'taille' éléments
    L = [randint(1,100) for i in range(1,taille)]

    # Relève le chronomètre
    temps_debut1 = time.time()
    # Boucle N fois le tri_selection L
    for i in range(N):
        tri_selection(L)
    # Relève le chronomètre
    temps_fin1 = time.time()
    temps1 = temps_fin1 - temps_debut1

    # Relève le chronomètre
    ...
    # Boucle N fois le tri_selection L
    ...
    # Relève le chronomètre
    ...

    return (temps1, ...)

t1, t2 = compare_les_tris(10000, 50)
print("Sur une liste de 50 éléments")
print("Temps pris par 10000 répétitions de tri_selection():",t1)
print("Temps pris par 10000 répétitions de sorted():",t2)
```

Commentaire :

On utilise à deux reprises ce schéma :

```
temps_debut = time.time()
programme_a_tester()
temps_fin = time.time()
temps = temps_fin - temps_debut
```

Une première fois pour chronométrer le `tri_selection()` et la seconde fois, pour chronométrer le tri de la liste par la fonction native `sorted()`

Nous avons déjà rencontré cette méthode dans l'activité r04e.