

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

Exercice : Comparaison graphique entre deux tris

Énoncé : Utilisation de la bibliothèque `pyplot` pour tracer l'évolution du temps d'exécution des appels en fonction de la taille de la liste.

```
# Importe la bibliothèque time et matplotlib.pyplot
from pylab import *

def compare_les_tris_et_trace(N:int):
    """ Lance N fois le tri d'une liste de taille croissante selon 2 algorithmes
        Mémoire pour chaque taille t de la liste, les temps d'exécution
        Trace le graphique d'évolution des temps de tri en fonction de la taille t de la liste
    """
    # Initialise les listes pour mémoriser la taille de L, le temps de chaque algorithme
    tailles = []
    temps_tri_selection = []
    temps_tri_sorted = []

    for t in range(10, 200, 10):
        temps1, temps2 = compare_les_tris(N, t)

        tailles.append(...)
        temps_tri_selection.append(...)
        temps_tri_sorted.append(...)

    plot(tailles, temps_tri_selection, label="tri_selection()")
    plot(tailles, ..., label="sorted()")
    xlabel("Taille des listes")
    ylabel("Temps")
    legend()

compare_les_tris_et_trace(50)
```

Commentaire :

On initialise les listes qui vont contenir les informations :

`tailles` : Tailles de listes

`temps_tri_selection` : Temps d'exécution du tri par sélection

`temps_tri_sorted` : Temps d'exécution du tri par `sorted()`

Pour des tailles différentes de listes, on chronomètre le temps d'exécution des deux algorithmes.

On les mémorise dans les listes ci-dessus, et on mémorise aussi la valeur de `t` dans la liste `tailles`

Finalement, avec `pyplot` on trace les graphiques de :

`temps_tri_selection` en fonction de `tailles`

et

`temps_tri_sorted` en fonction de `tailles`

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