

TP vitesse des tris

1 - Mise en place : *tris.py*

Récupérer les fichiers *tris.py*, *listes.py*, *temptri.py* et les placer dans un dossier *TPtris*. Ce dossier contient alors ces trois fichiers.

Ouvrir le fichier *tris.py*.

On a la fonction *tri_4* qui correspond au tri rapide.

Q1 : en identifiant avec les tris vus en classe, à quels tris correspondent *tri_1*, *tri_2* et *tri_3* ?

2 - Fichier *listes.py*

Dans ce fichier, on a 4 fonctions permettant de créer des listes ordonnées ou non.

Q2 : compléter la documentation de *cree_liste_decroissante* en donnant deux exemples.

On a besoin de créer des listes mélangées.

Q3 : compléter la fonction *cree_liste_melangee* afin d'obtenir une liste de nombres entiers compris entre 0 et n-1 mélangés.

Tester la fonction *cree_liste_melangee2*.

Q4 : quelle est la différence entre les fonctions *cree_liste_melangee* et *cree_liste_melangee2*?

3 - Le module **Pylab**

Le module *pylab* permet de fournir de nombreuses méthodes mathématiques et en particulier des outils pour tracer des graphiques (il intègre les fonctionnalités de *matplotlib* et *numpy*, on retrouve quasiment les mêmes instructions dans ces modules).

Tester le code suivant :

```
from pylab import *  
  
x = array([1, 3, 4, 6])  
y = array([2, 3, 5, 1])  
  
plot(x, y)  
  
show()
```

Q5 : détailler ce que font les instructions de ce programme.

On verra plus loin d'autres méthodes pour améliorer la présentation de ces graphiques.

4 - Le fichier *temptris.py*

a - La fonction **timeit**

Ouvrir ce fichier. La fonction `timeit` permet de mesurer le temps d'exécution d'une fonction (passée en paramètre) et réalisée un certain nombre de fois (ici 100 fois, on gardera ce paramètre à cette valeur dans ce TP).

Q6 : Tester la fonction `mesure_longueur(n, i)`. Comparer les valeurs obtenues avec $n=200$ et i entier de 1 à 4.

Q7 : Tester la fonction `mesure_longueur_max(n, i)`. Expliquer les différences avec la fonction précédente.

b - Des graphiques

On utilise `pylab` et on récupère les temps d'exécution avec la fonction précédente.

Pour tester rapidement (en patientant quand même) :

Q8 : tester la fonction `trace_1` sur un tableau créé avec `mesure_longueur_max(50, 1)`, enregistrer le résultat.

Afin d'avoir un peu plus de précisions :

Q9 : tester la fonction `trace_3` avec $n_{\max} = 100$ et les 4 tris (1 2 3 et 4), enregistrer les 4 images obtenues.

La fonction `trace_all()` permet d'avoir les graphiques simultanément du `tri_1` et du `tri_sort()`.

Q10 : modifier `trace_all()` pour avoir les graphiques des 5 tris avec $n_{\max} = 100$. Enregistre l'image et comparer les graphiques (performances, évolutions ...).