

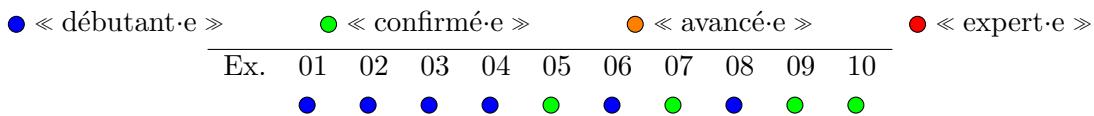
Sujet n°5

Algorithmique et Structure de Données Algorithmes de tri

Temps de réalisation: 1h30mn

L'objectif de ce TP est de mettre en pratique les algorithmes de tri.

Échelle de progression :



Pour acquérir un niveau de compétence donné, il fautachever **tous** les exercices de ce niveau et des niveaux inférieurs. L'objectif est d'atteindre le niveau *confirmé·e* (●) à l'issue du TP.

Table des matières

1 Mise en place	1
Exercice 1 : Génération de 6 listes	2
Exercice 2 : Calcul complexité en temps	2
Exercice 3 : Tableau comparatif	2
2 Les algorithmes de tri en $O(n^2)$	2
Exercice 8 : Implémentation	2
Exercice 9 : Calcul du temps de réponse	2
3 Les algorithmes de tri en $O(n \log_2 n)$	2
Exercice 8 : Implémentation	2
Exercice 9 : Calcul du temps de réponse	2
4 Les algorithmes de tri en $O(n)$	3
Exercice 8 : Implémentation	3
Exercice 9 : Calcul du temps de réponse	3
5 Interprétation des résultats	3
Exercice 10 : Interprétation des résultats	3

1 Mise en place

Dans ce TP, vous allez tester tous les algorithmes de tri que nous avons vu en cours.

Nous souhaitons dans ce TP, en plus de tester les algorithmes de tri sur certaines données, de comparer les résultats obtenus sur les mêmes données.

Ainsi, vous devez dans la suite réaliser les étapes présentées ci-dessous.

Exercice 1: Génération de 6 listes

Générer 6 listes de 10, 100, 1000, 10000, 100000 et 1000000 éléments entiers. Les éléments seront générés de manière aléatoire en utilisant la bibliothèque *random*.

```
import random

n = random.randint(0, 100000)
print(n)
```

Exercice 2: Calcul complexité en temps

Mettez en place une architecture pour pouvoir calculer la complexité en temps de chacun des algorithmes que vous allez développer dans la suite. Vous allez exécuter ces algorithmes sur les 6 listes que vous avez générées auparavant.

Le code ci-dessous permet de calculer le temps d'exécution de l'algorithme ALGO executé sur la liste LISTE.

```
import time

start_time = time.time()
result = ALGO_TRI(LISTE)
end_time = time.time()
print(f"Complexité en temps d'exécution: {end_time - start_time} seconds")
```

Exercice 3: Tableau comparatif

Mettez en place un tableau comparatif qui prend sur les lignes les 6 listes et sur les colonnes les différents algorithmes de tri. Les valeurs de ce tableau correspondent au temps d'exécution de chaque algorithme sur chaque liste.

2 Les algorithmes de tri en $O(n^2)$

Exercice 4: Implémentation

Implémentez les algorithmes de tri en $O(n^2)$ vus en cours.

Exercice 5: Calcul du temps de réponse

Calculer le temps de réponse des algorithmes implémentés et ajouter les valeurs dans le tableau créé auparavant.

3 Les algorithmes de tri en $O(n \log_2 n)$

Exercice 6: Implémentation

Implémentez les algorithmes de tri en $O(n \log_2 n)$ vus en cours.

Exercice 7: Calcul du temps de réponse

Calculer le temps de réponse des algorithmes implémentés et ajouter les valeurs dans le tableau créé auparavant.

4 Les algorithmes de tri en $O(n)$

Exercice 8: Implémentation

Implémentez les algorithmes de tri en $O(n)$ vus en cours.

Exercice 9: Calcul du temps de réponse

Calculer le temps de réponse des algorithmes implémentés et ajouter les valeurs dans le tableau créé auparavant.

5 Interpretation des résultats

Exercice 10: Interpretation des résultats

A partir du tableau des temps de réponse, donnez votre interprétation par rapport aux algorithmes de tri implémentés.