

Algorithmes et structures de données avancées : TD 1(sur machine)

Listes en Python - Temps d'exécution

Pour initialiser des listes avec n éléments, vous disposez des deux manières suivantes :

```
l = [ 0 for i in range(n) ]  
# ou bien  
l = [0] * n
```

Pour chronométrer la durée d'une fonction, vous disposez du code suivant :

```
import time  
tempsDebut = time.time()  
  
#ici votre algorithme (affectations, boucles, conditions, appels de fonctions, ...)  
  
tempsFin = time.time()  
print 'Temps d exécution : %0.3f s' % ((tempsFin-tempsDebut)*1.0)
```

Exercice 1.1 *Les listes (liste à une dimension)*

1. Créer une liste `liste` avec 10 éléments et afficher la liste.
2. Changer chaque élément de la liste pour qu'il contienne une valeur aléatoire entre 1 et 15 (*Remarque : Utiliser la fonction `random.randint(inf, sup)` après avoir chargé le module correspondant avec `import random`*).
3. Afficher la nouvelle liste.
4. Ecrire un algorithme qui compte le nombre d'occurrences de la valeur 10 dans la liste.

Exercice 1.2 *Temps d'exécution d'algorithmes sur les listes*

Dans cet exercice, nous allons analyser les temps d'exécutions de vos algorithmes pour les listes.

1. Ecrire un programme en python qui demande à l'utilisateur de saisir une valeur numérique qui sera stocké dans la variable `n`.
2. Créer une liste `liste` vide avec n éléments.
3. Changer chacun des n éléments de la liste pour qu'il contienne une valeur aléatoire entre 1 et 15 (*Remarque : Utiliser la fonction `random.randint(inf, sup)` après avoir chargé le module correspondant avec `import random`*).
4. Ecrire un algorithme qui compte le nombre d'occurrences de la valeur 10 dans la liste, et qui affiche le nombre d'occurrences à l'écran.

5. Chronométrer le temps d'exécution de votre algorithme à l'aide du code fourni en haut de la page, pour une saisie de valeur pour $n=10$, $n=100$, $n=1000$, $n=10000$, $n=100000$, $n=1000000$, $n=10000000$, $n=100000000$. *Remarque : Si votre logiciel ne répond plus après un certain temps, vous pouvez appuyer sur la combinaison de touche CTRL-C.*
6. Tracer un graphe en fonction de n pour le temps d'exécution avec un logiciel de tableur (p.ex. Excel).

Exercice 1.3 *Temps d'exécution de fonctions prédéfinies sur les listes*

1. Dans votre code précédent, remplacer votre algorithme qui compte les occurrences de la valeur 10 dans la liste par la fonction prédéfini `count`.
2. Chronométrer de nouveau le temps d'exécution pour une saisie de valeur pour $n=10$, $n=100$, $n=1000$, $n=10000$, $n=100000$, $n=1000000$, $n=10000000$, $n=100000000$.
3. Tracer un graphe en fonction de n pour le temps d'exécution avec un logiciel de tableur (p.ex. Excel).