

Python insia TP 2

Objectifs du TP

Le principe du TP va être de comparer les performances de plusieurs versions de la solution au dernier exercice du TP sur la génération de phrases, à l'aide de décorateurs.

Les objectifs sont les suivants :

- Découvrir les décorateurs ;
- Avoir quelques notions sur l'optimisation et la mesure des performances ;
- Voir l'importance des caches ;
- Voir l'importance des algorithmes et structures de données.

Plan du TP

1. Présentation du programme `phrase-raw` et Utilisation du programme avec différents paramètres ;
2. Présentation des décorateurs ;
3. Écriture d'un décorateur de profiling simple : il doit compter dans un dictionnaire le nombre de fois qu'une fonction est appelée, et à la fin du programme, on affiche ce dictionnaire ;
4. Amélioration du décorateur : compter aussi le temps écoulé dans la fonction, et afficher à la fin, classer par temps écoulé, la fonction, le nombre d'appels, le temps total et le temps moyen ;
5. Application sur le programme `phrase-new` pour déterminer ce qui prend du temps ;
6. Digression sur Fibonacci : rappels de la suite de Fibonacci, implémentation naïve ;
7. Décorateur de cache, et application à la suite de Fibonacci ;
8. Version améliorée du programme initial, avec un cache : `phrase-cache` ;
9. Étude des performances de `phrase-cache` ;
10. Seconde version améliorée : `phrase-first` ;
11. Étude des performances de `phrase-first` ;
12. Version optimale: `phrase-smart`.

Remarques sur le tableau de résultats

- L'utilisation d'aléatoire dans le programme rend tout bench difficile ;
- Même fixer le générateur ne suffit pas, car différents algorithmes vont avoir différents résultats (en particulier parce que les tables de hashage n'ont pas d'ordre) ;
- La version `phrase-smart` a un temps quasiment constant, la construction de la structure de données prend le temps, l'utilisation est quasi-instantanée, ce genre de situations est souvent optimal.