

SAE 1.02 Comparaison d'approches algorithmiques

Dans le main, on crée une variable de début de temps, une de fin de temps et 3 variables de temps total (une pour chaque tri), 2 tableaux d'entiers de 100 valeurs aléatoires, nous ferons un tableau que nous copierons dans le deuxième entre chaque appel et nous ne modifierons que ce deuxième tableau. Ainsi chaque tri aura les mêmes valeurs à trier. On les testera 20 fois avec un nouveau tableau. Puis à la fin, on calculera une moyenne du temps de chacun pour comparer. On répète ensuite l'opération avec un autre tableau de 500 valeurs, puis 1000 valeurs et enfin 10 000 valeurs

Afin de calculer chaque moyenne de temps pour chaque appel, on lance un chrono au début de chaque tri qui nous retourne le temps pris par le tri. Nous ajouterons ce temps à la variable totale attribuée à ce tri à chaque boucle puis une fois le tout terminé, nous diviserons par le nombre de fois où nous sommes entrés dans la boucle (ici 20 fois) afin d'obtenir la moyenne de chaque temps.

L'utilisation d'entiers permet d'améliorer considérablement le temps d'exécution.

- Tri à bulle :

Le tri à bulle consiste à faire remonter le plus grand élément du tableau en comparant les éléments successifs. On va donc, par exemple, comparer les deux premières valeurs du tableau puis les inverser si la première valeur est plus grande que la deuxième. Ce tri va utiliser deux boucles, une pour répéter le processus, elle sera représentée par un cpt2 qui va augmenter jusqu'à qu'il soit égal à la taille - 1. L'autre boucle, elle, va parcourir le tableau pour trier les valeurs, et s'arrêtera quand cpt1 est égal à la taille - cpt2 puisqu'on n'a pas besoin de parcourir la dernière valeur alors qu'elles sont triées.

- Tri par sélection :

Pour réaliser ce tri, on initialise un tableau d'entiers. On déclare et on initialise une variable $cpt = 0$,

Si la valeur de la dernière case est plus petite que n'importe quelle autre valeur du tableau, la plus grande valeur se retrouve placée à la dernière case. Puis cpt est incrémenté de 1 à la fin de chaque action.

On répète l'opération jusqu'à ce que toutes les valeurs soient triées. ($cpt > \text{taille}$)

- Tri par insertion :

Afin de réaliser le tri par insertion, nous avons besoin d'un tableau d'entiers entré en argument ainsi que de deux variables cpt: cpt1 et cpt2.

Nous aurons deux boucles tant que: la première qui se fera tant que cpt1 sera inférieur à la taille du tableau et qui commencera à 1 (la première valeur sera considérée comme "déjà triée")

Dans cette boucle nous initialiserons notre cpt2 à cpt1 -1 afin de comparer la valeur de cpt1 dans le tableau à celle qui la précède. Si cette valeur est plus grande on l'échange et on diminue cpt2 de 1 jusqu'à ce que la valeur de cpt2 soit inférieure à la valeur de cpt2 +1 ou qu'il n'y aie plus de valeur à comparer avant (donc que cpt2 soit inférieur à 0). En revanche, si la valeur de cpt1 est plus grande que celle de cpt2, nous n'aurons pas besoin de l'échanger et nous pourrons directement continuer dans notre tableau.

On répète cette opération jusqu'à ce que cpt1 soit égal à la taille du tableau en n'oubliant pas d'incrémenter cpt1 de 1 à chaque boucle et de décrémenter cpt2 tant qu'il en aura besoin dans sa propre boucle et nous devrions avoir un tableau trié.