Compte rendu - Projet programmation fonctionnelle

Dorine Descamps et Marwan Ait Addi

27 avril 2019

Table des matières

Afin de faire un choix de liste cohérent, nous avons testé le temps que mettaient nos fonctions à trier des listes en faisant changer plusieurs variables. L'ordre étant une variable possible nous gardons l'ordre croissant pour tous les autres tests. Nous arrondirons les temps en fonctions des résultats, l'unité étant la seconde.

1 Comparaison du temps des fonctions de tri en augmentant la taille des listes

Nous nous plaçons dans une fourchette d'entier de 0 à 100 pour le choix des éléments de la liste et augmentons la taille de la liste en multipliant par 10 à chaque fois.

nombre d'éléments de la liste	10	100	1000	10 000
temps tri_partition_fusion	0	0	0.001	0.041
temps tri_pivot	0	0.001	0.002	0.065
temps tri_bulle	0	0.002	0.2	28.34

Il apparaît déjà que le tri à bulle est la manière de trier la moins optimisé sur de grande liste.

2 Comparaison du temps des fonctions de tri en augmentant la fourchette d'éléments possible

Cette fois-ci, nous gardons la taille de la liste constante à 1000 éléments et augmentons la fourchette d'entier possible.

fourchette d'éléments de la liste	50	500	5000	10 000
temps tri_partition_fusion	0.003	0.003	0.003	0.006
temps tri_pivot	0.002	0.002	0.002	0.004
temps tri_bulle	0.22	0.25	0.23	0.25

Nous pouvons remarqué que la fourchette d'éléments n'est pas une variable très important, les résultats sont relativement stables.

3 Comparaison du temps des fonctions de tri en fonction de l'ordre

Pour cette partie nous prenons une liste de 1000 éléments compris entre 0 et 5000 et changeons l'ordre de tri.

ordre	<	>	<=	>=
temps tri_partition_fusion	0.004	0.003	0.0030	0.002
temps tri_pivot	0.002	0.002	0.0020	0.001
temps tri_bulle	0.23	0.21	0.229	0.23

Là aussi, nous remarquons que l'ordre n'impacte pas beaucoup le temps de résolution, les résultats restent plutôt constants.

4 Comparaison du temps des fonctions de tri en fonction du nombre de doublons

Ici, nous utilisons des listes de 100 éléments modifiés à la main pour avoir la quantité de doublons voulue.

Quantité de doublons	100%	environ 50%	25%	aucun
temps tri_partition_fusion	0.001	0	0.001	0.001
temps tri_pivot	0.08	0	0.001	0
temps tri_bulle	0	0.004	0.005	0.001

Les chiffres de cette expérience ne donnent pas de résultats flagrants, si ce n'est que le tri à bulle est toujours à la traine.

5 Comparaison du temps des fonctions de tri avec des listes déjà triées