Projet sur la complexité des algorithmes de tri

Aziz M'HIRSI

Université du Havre

Consigne

Comparaison des algorithmes de tri : par insertion, fusion, dénombrement. Le tri s'effectue sur un tableau d'éléments. Les éléments comportent un champ clé (entier) et un champ valeur (chaîne de caractères). Les clés peuvent être supposées uniques dans un premier temps.

Votre programme prendra en entrée le tableau, qui pourra avoir été généré aléatoirement (pour tester des tableaux de grande taille) et éventuellement lu à partir d'un fichier.

Les comparaisons entre les tris se feront grâce à un comptage des opérations élémentaires : comparaison de clés, affectation (clés ou éléments), et par la récupération du temps d'exécution.

Vous devrez rendre une présentation courte de chaque tri, le code java, et des courbes, que vous commenterez, du temps d'exécution en fonction de la taille du tableau.

Projet sur la complexité des algorithmes de tri

Le but de l'exercice est de tester les différent type de tri sous le langage Java. Ici, les tris tester son le Tri par Insertion, Tri par Fusion et Tri par Dénombrement. Les critères de comparaison sont : le temps d'exécution, le nombre de calcul élémentaire (nombre incrémentation et de comparaison de clés) en fonction du nombres d'élément a trier.

Méthode

Le tri par insertion

L'algorithme de tri va trier au fur et à mesure qu'il lit les éléments à trier. En effet, tous les éléments avant celui-ci sont triés. Le but est que l'élément lu sera mis à sa place. Ainsi, l'algorithme va comparer l'élément lu avec les éléments qui le précède jusqu'à trouver sa place. Une fois la place trouvée, il ne reste plus qu'à déplacer les éléments supérieurs pour insérer notre élément.

Le tri par insertion a une en moyenne, pour 10 élément, avec des clés inférieures à 100, un temps d'exécution de 136.9 ms, 40.6 affectation et 22.6 comparaison de clés.

Le tri par fusion

L'algorithme de tri est un algorithme récursif, c'est-à-dire qui fait appelle à lui-même. L'algorithme va séparer en deux la liste des éléments jusqu'à obtenir un élément seul. Une fois que cet élément et seul on va regarder si la deux partie est elle aussi un élément seul. Si c'est le cas, on fusionne les deux liste et on les tris. Sion on divise à nouveau la deuxième liste. Et continue jusqu'à obtenir une liste triée.

Le tri par fusion a une en moyenne, pour 10 élément, avec des clés inférieures à 100, un temps d'exécution de 131.6 ms, 48.3 affectation et 22.3 comparaison de clés.

Le tri par dénombrement

L'algorithme de tri va créer un tableau de comptage dans lequel il va lister le nombre de clés du tableau a trié par numéro de clés puis venir mettre les éléments dans l'ordre des numéros de clés en comparant avec le tableau a trier et dans un nouveau tableau qui sera trié.

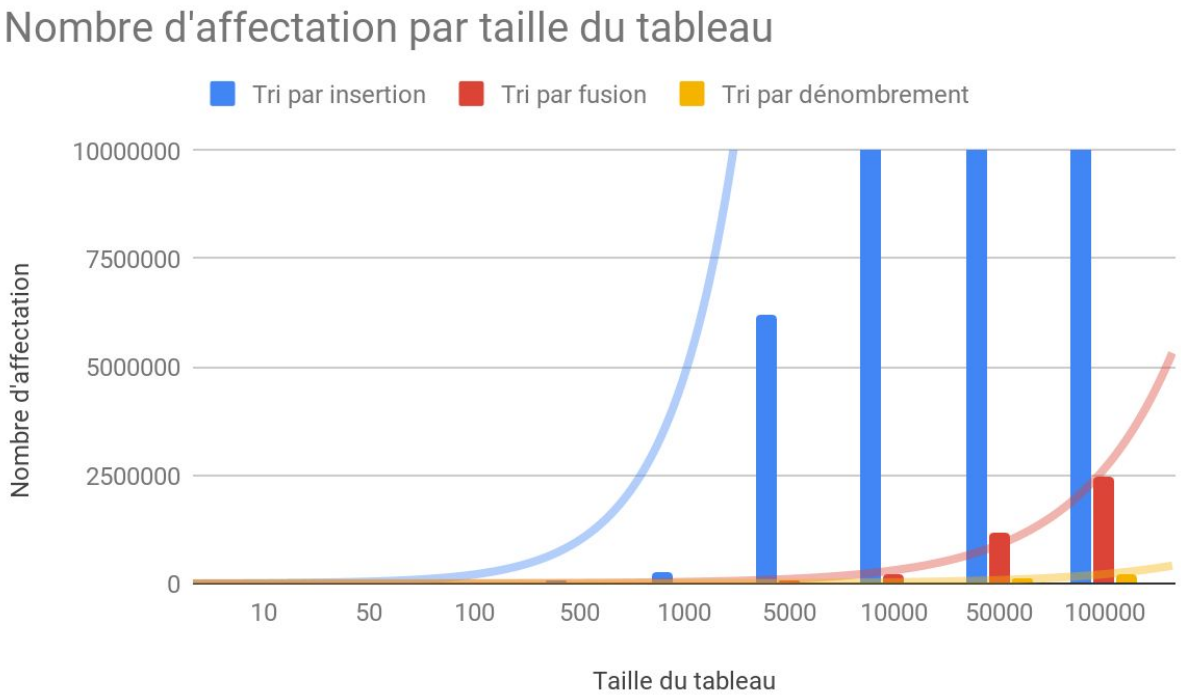
Le tri par fusion a une en moyenne, pour 10 élément, avec des clés inférieures à 100, un temps d'exécution de 130.7 ms, 20 affectation et 0 comparaison de clés.

Résultats

Pour nos diagrammes à colonnes, on pose la valeur maximale des clés (borneMax) à 100 000 dans un premier temps. Puis on posera le tableau à une taille de 100 élément (nb) et on fera varier la valeur maximale..

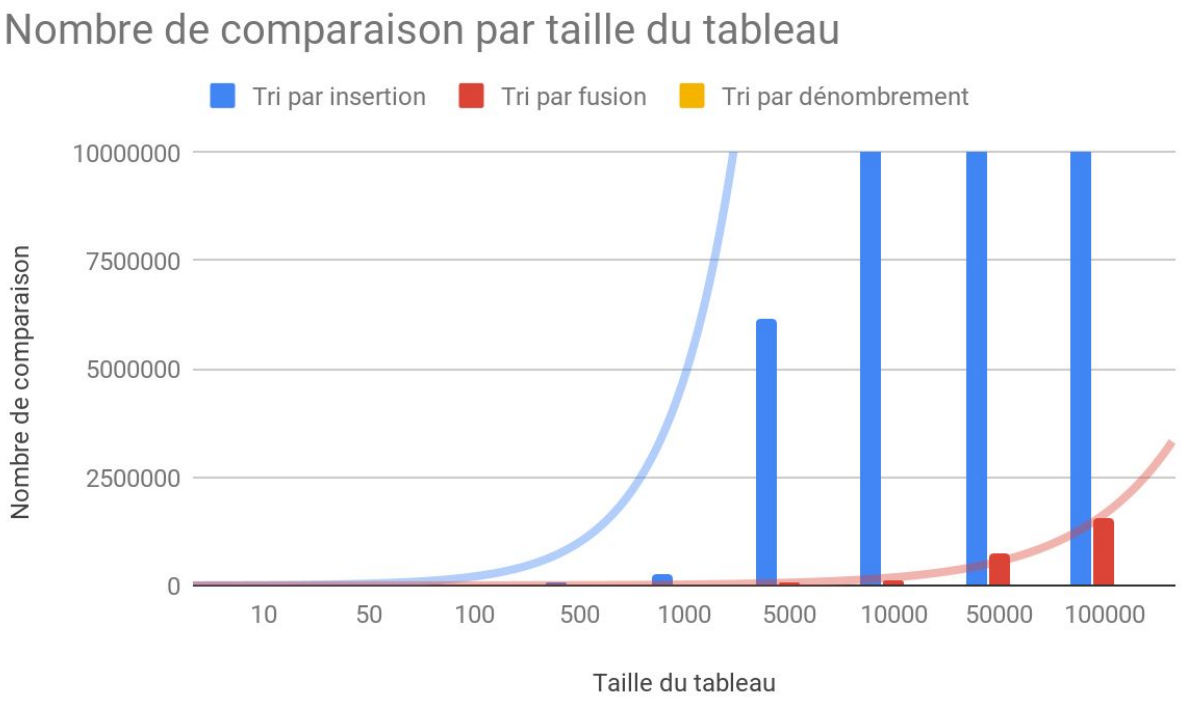
Pour une meilleure lisibilité, on limitera les tableaux à 10 000 000 opérations élémentaire.

Nombre d'affectation par taille du tableau



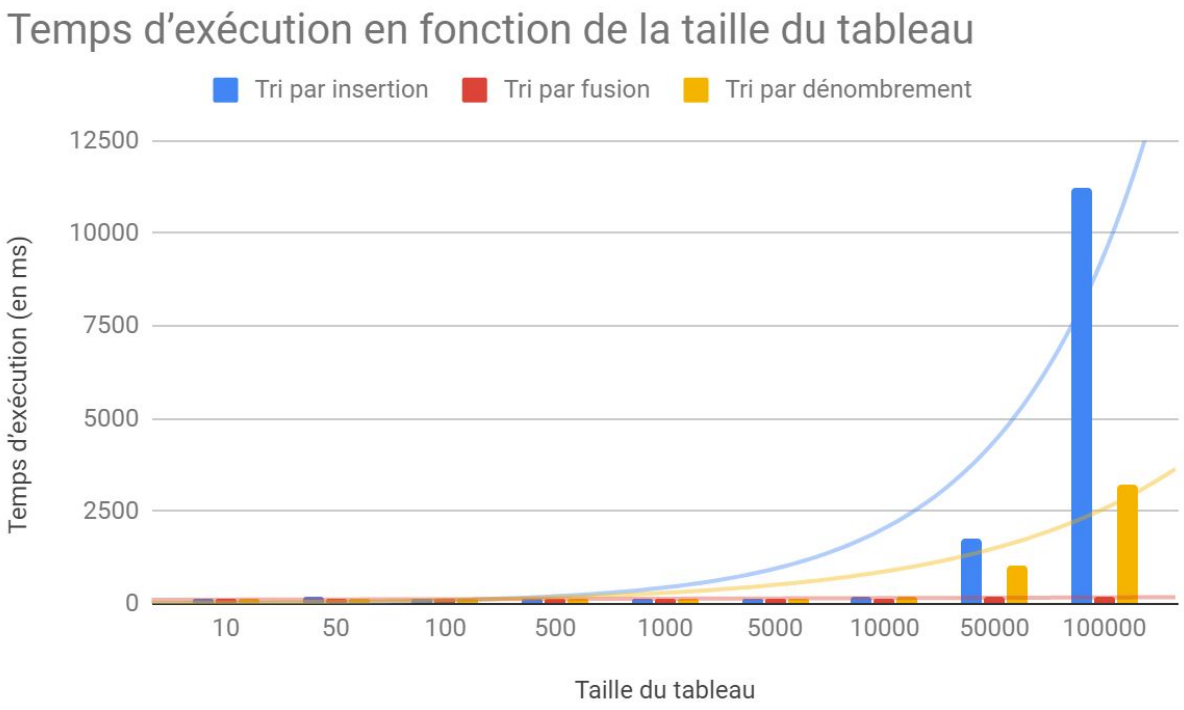
Grâce au diagramme, on voit que le tri le plus coûteux est le tri par insertion avec ses 2 506 528 147 affectation pour 100 000 éléments suivi par le tri fusion qui est bien plus efficace avec 2 467 614 affectation pour 100 000 éléments, suivi de près par le tri par dénombrement avec seulement 200 000 affectation pour 100 000 éléments.

Nombre de comparaison de clés par taille du tableau



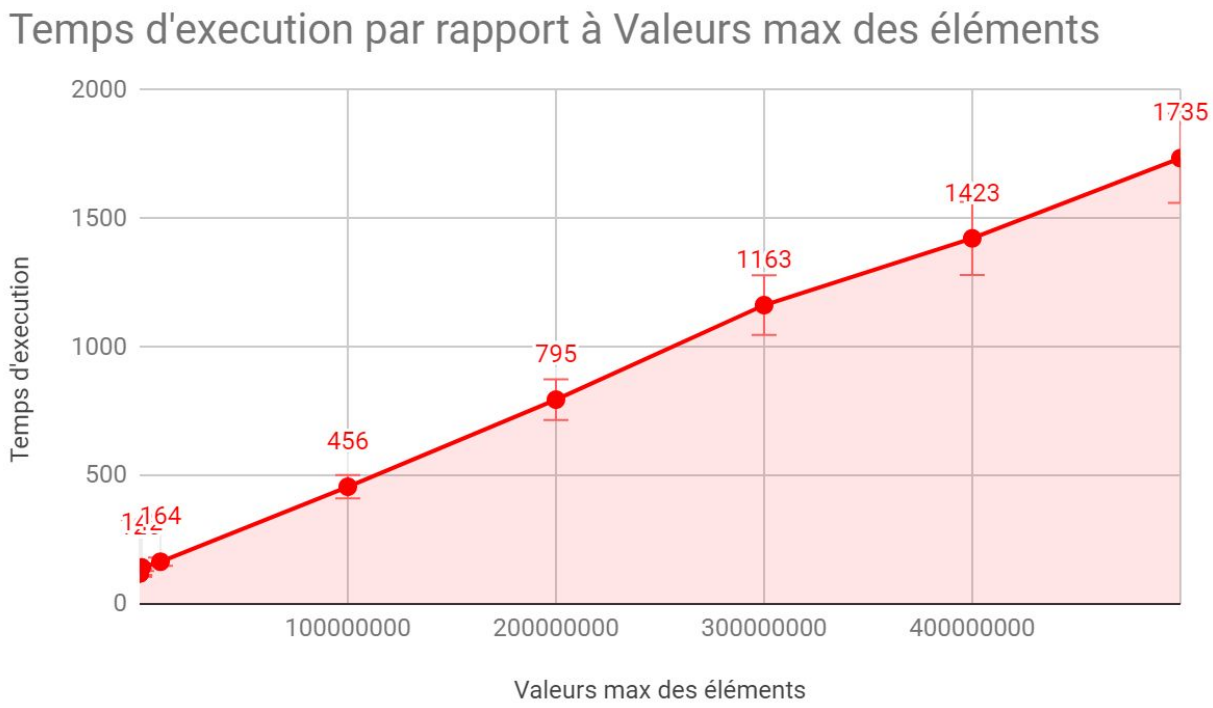
On remarque que le tri par dénombrement est le plus efficace en nombre de comparaisons de clé, car il n'y en a aucune. Mais comme on peut le voir sur la courbe de tendance, le tri par insertion explose les plafonds. Pour un tableau de 100 000 éléments un total de 2 506 328 149 comparaison de clés, par rapport au 1 536 341 comparaison de clés pour le tri par fusion.

Temps d'exécution par taille du tableau



Au niveau des temps d'exécution, le tri par fusion est le plus rapide avec seulement 189,5 ms d'exécutions pour un tableau de 100 000 éléments ce qui est très rapide. Le tri par dénombrement est plus lent, mais reste avec un temps honorable de 3. 213 s d'exécution. Le tri par insertion est lui à la traîne avec 11. 253 s de temps d'exécution pour le mm tableau

Temps d'exécution par rapport à Valeurs max des éléments pour le tri par dénombrement



On voit que plus la taille du tableau de comptage est grand plus le temps d'exécution sera important donc il ne faut pas créer de trop grand tableau de comptage pour n'en utiliser qu'une petite partie. En plus de prendre du temps, mais aussi notre précieuse mémoire.

Conclusion

On observe que notre grand vainqueur est le tri par dénombrement qui est le plus efficace en générale. C'est le moins coûteux en opération élémentaire. Mais pour obtenir des temps d'exécution plus court le tri par fusion est le meilleur compromis. Le tri par insertion quant à lui n'est pas du tout efficace.