Participants ASSANI Said 22001535, ABBAS Ali 22001535

Rapport du TP noté en Algorithmes avancée : Les algorithmes de tri sur les listes chainées en Java.

Table des matières

Problématique : Quel fonction de tri est le mieux adapter pour le rangement croissant d'une liste chainée ?

Introduction

Nous allons rechercher qu'elle fonction de tri parmi le tri par insertion, le tri à bulle, le tri rapide et le tri par fusion est le mieux adapter pour ranger dans l'ordre croissant la liste chainée étudié en cours.

Pour cela nous avons lancer les fonctions de tri sur plusieurs listes générées aléatoirement en faisant varier la taille de la liste, l'étendu, la médiane, l'écart type, la variance et d'autres critères, mais aussi en soumettant le tri à différents cas, le cas « générale » sur laquelle les éléments sont positionner aléatoirement, et les 2 pires cas possibles, lorsque la liste est déjà rangée et lorsque la liste est ranger dans l'ordre inverse.

Implémentation des fonctions et méthodes

Pour ne pas biaiser les résultats, nous avons construit chacune des méthodes de tri (par ordre croissant) avec les algorithmes qui nous semblaient les plus rapides possible.

Contrairement aux fonctions de tri, le reste des méthodes (comme pour exporter les données, calcul de moyenne, médiane, copie de liste etc.) n'ont pas été conceptualisé pour avoir la meilleur compléxité en temps possible. Malgré tout il n'influe pas sur la mesure de temps puisque le temps est mesuré exactement avant et exactement apres l'éxécution de la fonction de tri seul.

Environnement de reccueil des données

Lors de l'exécution du projet, seul Intellij et l'explorateur de fichiers Windows était en cours d'exécution au premier plan. Le projet a été exécuter plusieurs fois à des jours d'intervalles et sur différents os (MacOS et Windows), les résultats étaient semblables aux exécutions précédentes.

Hypothèses de recherches

D'après Wikipédia : (avec n étant la longueur de la liste)

Tableau comparatif des tris procédant par comparaisons

Nom	Cas optimal	Cas moyen	Pire des cas	Complexité spatiale	Stable
Tri rapide	$n \log n$	$n \log n$	n^2	$\log n$ en moyenne, n dans le pire des cas ; variante de Sedgewick : $\log n$ dans le pire des cas	Non
Tri fusion	$n \log n$	$n \log n$	$n \log n$	n	Oui
Tri par tas	$n \log n$	$n \log n$	$n \log n$	1	Non
Tri par insertion	n	n^2	n^2	1	Oui
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	Non
Tri par sélection	n^2	n^2	n^2	1	Non
Timsort	n	$n \log n$	$n \log n$	n	Oui
Tri de Shell	n	$n \log^2 n$ ou $n^{3/2}$	$n \log^2 n$pour la meilleure suite d'espacements connue	1	Non
Tri à bulles	n	n^2	n^2	1	Oui
Tri arborescent	$n \log n$	$n \log n$	$n \log n$ (arbre équilibré)	n	Oui
Smoothsort	n	$n \log n$	$n \log n$	1	Non
Tri cocktail	n	n^2	n^2	1	Oui
Tri à peigne	n	$n \log n$	n^2	1	Non
Tri pair-impair	n	n^2	n^2	1	Oui

- En général dans le meilleur des cas, le tri rapide et le tri par fusion sont les meilleurs avec une complexité de n log(n) suivit du tri à bulles et le tri par insertion avec une complexité de n. (tri rapide >= tri par fusion >tri à bulles >= tri par insertion)
- Dans le pire des cas, le tri par fusion est le meilleur avec une complexité de n log n, suivit des 3 autres avec une complexité de n^2. (Tri par fusion > tri rapide >= tri à bulles >= tri par insertion)

Analyse

Pour ne pas encombré le rapport nous avons décidé d'exposer que les graphiques, mais vous pouvez accéder aux détails des listes générés (tailles, modes, étendues, médianes, écart type, moyennes etc.) en vous référant au fichier .csv correspondant dans ./data/ ou aux images dans ./data/graphique.

Liste Quelconque

Résultats

Figure 1 : Liste quelconque avec des éléments allant de 0 à 10.

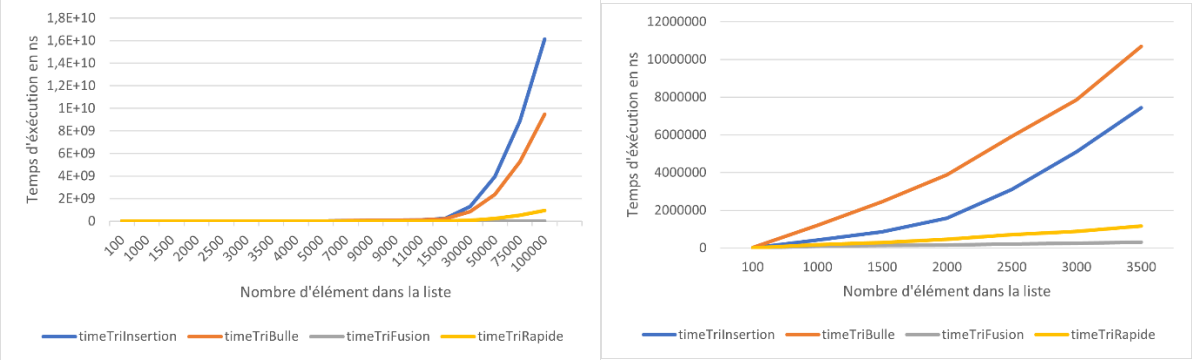

Figure 2 : Zoom de 0 à 3500 sur Figure 1.

Figure 3 : Liste quelconque avec des éléments allant de 0 à 1000.

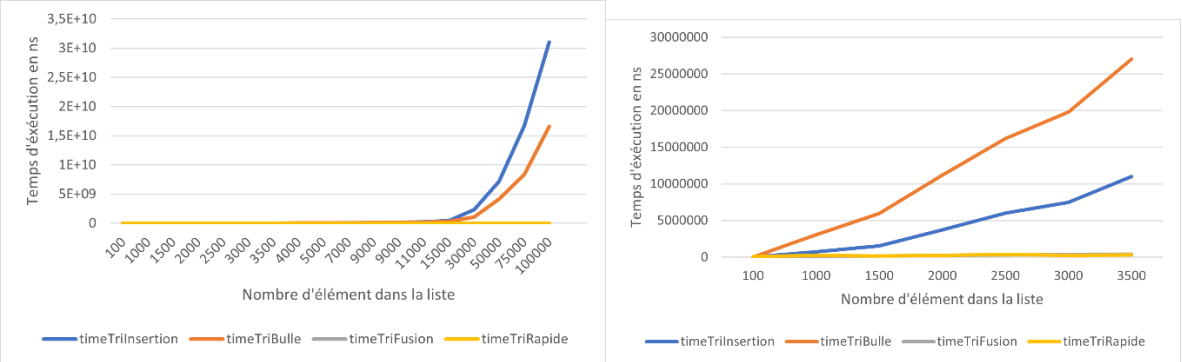

Figure 4 : Zoom de 0 à 3500 sur figure 3.

Figure 5 : Liste quelconque avec des éléments allant de 0 à 100 000.

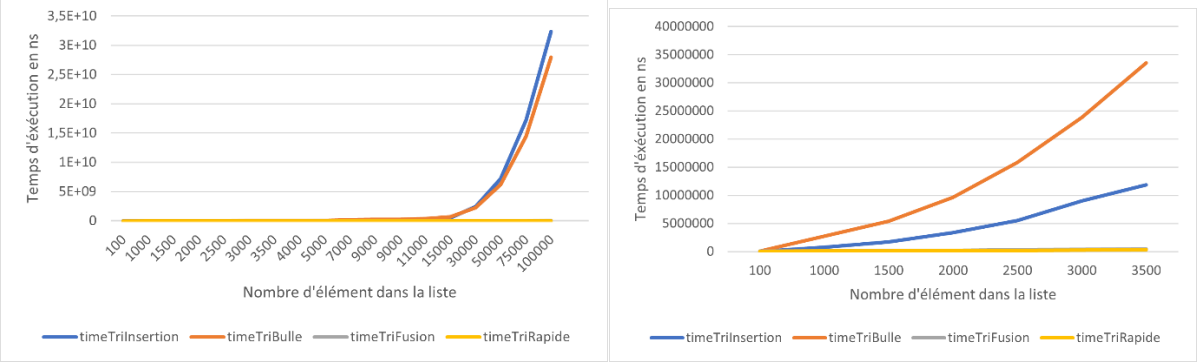

Figure 6 : Zoom de 0 à 3500 sur figure 5.

Interprétations

Nous pouvons voir qu'incontestablement sur les 3 cas le tri par fusion et le tri rapide sont de loin les plus rapide. Bien que les temps se rapprochent extrêmement le plus rapide est le tri par fusion, cela est nettement visible lorsque l'écart type et la variance est très faible, plus le nombre d'élément dans la liste augmente et plus la différence est prononcée. Contrairement à la figure 5 qui est le graphique auquel l'écart type et la variance sont le plus élevés nous voyons que bien que le nombre d'élément de la liste augmente la courbe de tri par fusion et tri rapide restent confondu du début à la fin.

A la comparaison le tri par insertion et le tri à bulle sont très lent. Jusqu'à environs 10 000 éléments, le tri par insertion est meilleur que le tri à bulles, puis au-delà le tri par insertion devient beaucoup plus lent.

A noter que plus l'écart types et la variance augmentent et plus les 2 tris ralentissent drastiquement.

Liste Croissante

Résultats

Figure 7 : Liste croissante avec des éléments allant de 0 à 10.

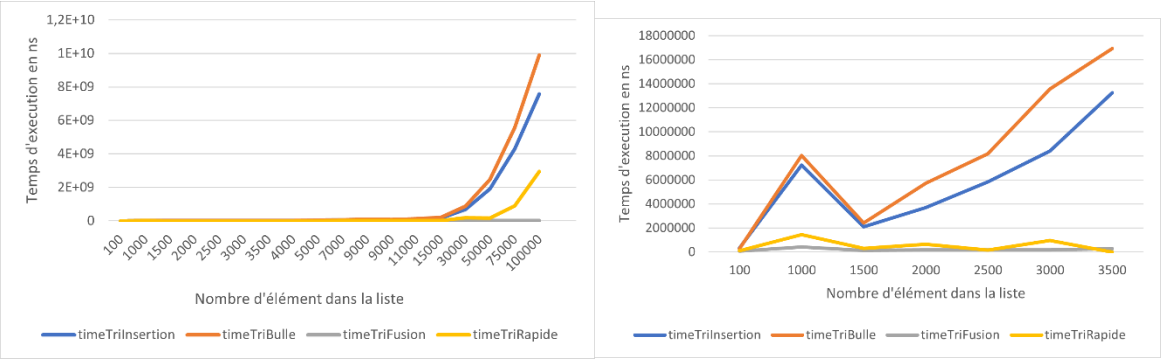

Figure 8 : Zoom de 0 à 3500 sur figure 7.

Figure 9 : Liste croissante avec des éléments allant de 0 à 1000.

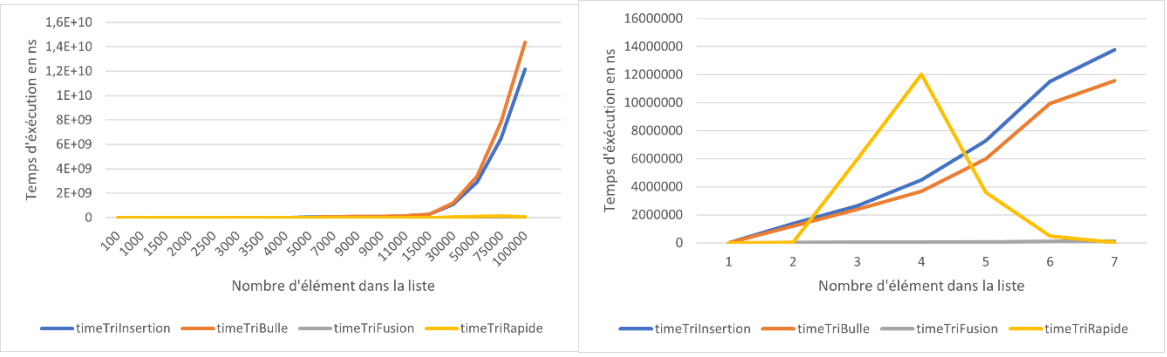

Figure 10 : Zoom de 0 à 3500 sur figure 9.

Figure 11 : Liste croissante avec des éléments allant de 0 à 100 000.

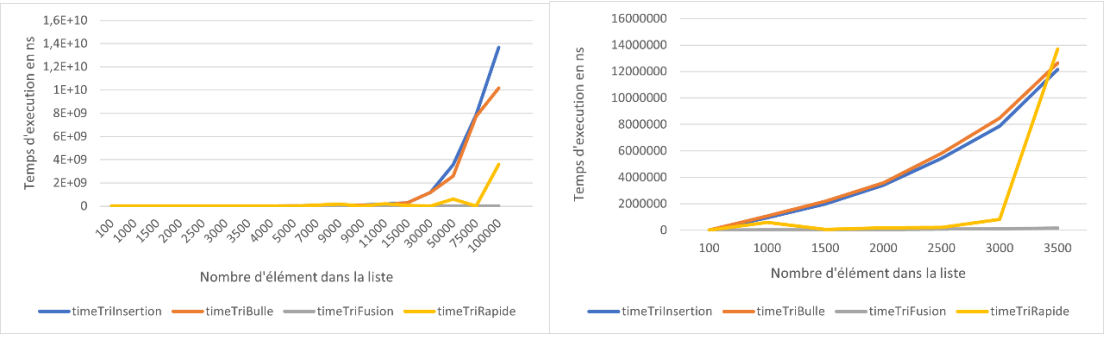

Figure 12 : Zoom de 0 à 3500 sur figure 11.

Interprétations

Les positions de vitesse n'ont pas changé, le tri par fusion reste le meilleur, suivit du tri rapide, du tri à bulle et le tri par insertion. La grande différence se trouve sur la vitesse d'exécution de tous les tris qui se sont vu beaucoup diminuer comparer aux tris sur les listes quelconques. Nous pouvons déduire que plus la liste d'origine est rangée et plus se sera rapide de trier l'entièreté de la liste. A noter qu'en dessous de 75000 éléments, plus l'écart type et la variance sont faible et plus le tri à bulle devient lent.

Liste Décroissante

Résultats

Figure 13 : Liste décroissante avec des éléments allant de 0 à 100.

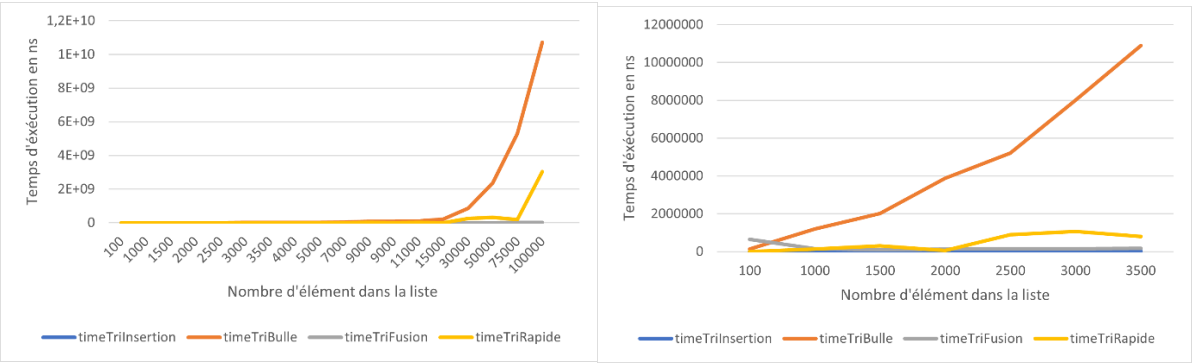

Figure 14 : Zoom de 0 à 3500 sur figure 13.

Figure 15 : Liste décroissante avec des éléments allant de 0 à 1000.

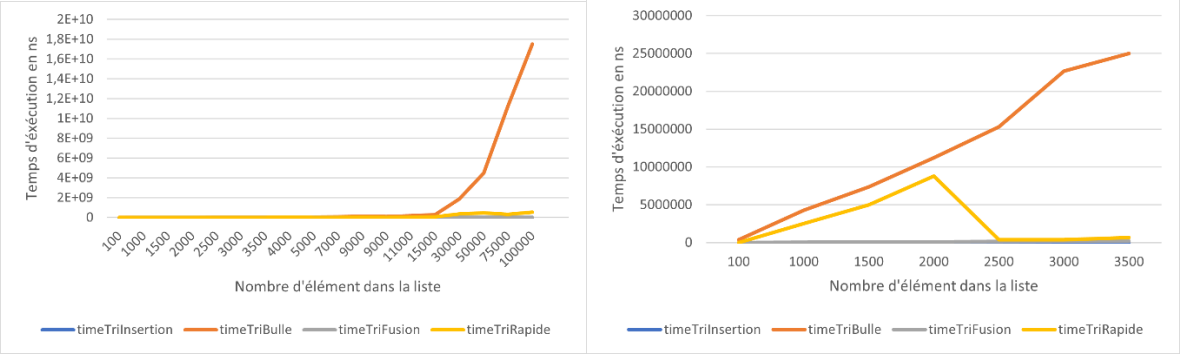

Figure 16 : Zoom de 0 à 3500 sur figure 15.

Figure 17 : Liste décroissante avec des éléments allant de 0 à 100 000.

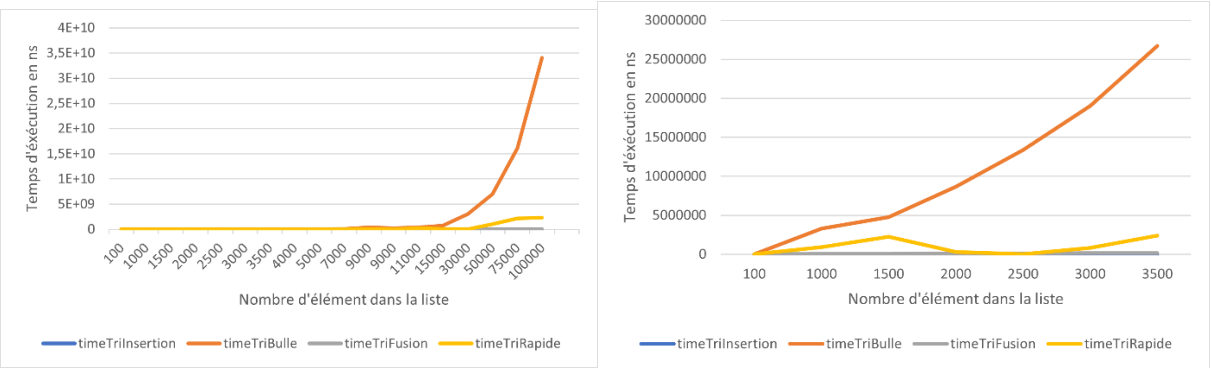

Figure 18 : Zoom de 0 à 3500 sur figure 17.

Interprétations

Le tri à bulle est très lent, cela peut se voir sur la figure 18, des le départ pour des « petites » listes de 1000 à 3000 éléments, sont temps de résolution augmentent très vites. Le tri rapide lui oscille en augmentant et en diminuant encore et encore. Plus la variance et l'écart type augmente plus le tri à bulle et le tri rapide augmentent au niveau de la lenteur.

Le tri par insertion et le tri par fusion sont eux très rapide, peux importe la variance et la taille de la liste.

Conclusions

Le tri par fusion est excellent dans tous les cas, il occupe la première place sur toutes les sortes de listes rangées en croissant, décroissant, quelconque, et indépendamment de la taille de la liste de l'écart type, la variance etc.

Le tri rapide est bien aussi, elle est très rapide sur les listes quelconques mais il ralentit lorsque la liste est déjà triée en croissant et que la variance augmente également il ralentit sur les listes trier en décroissant lorsque la variance est très basse.

Contrairement au tri par insertion qui a été la plus lente sur les listes quelconques et les listes déjà trier en ordre croissantes, mais est étonnamment le plus rapide sur les listes rangées en ordre décroissant.

Le tri à bulle est le plus lent de tous, il fait partie des derniers sur tous les cas, à aucun moment il ne fait partie des 2 plus rapides, mais aussi c'est celui qui cumule le temps le plus élevé et qui a le temps le plus élevé toutes mesurent confondus qui est lorsque la liste est décroissante, plus la liste est longue et la variance est élevé et plus il ralentit.

Nous nous retrouvons avec l'ordre « tri fusion > tri rapide> le tri insertion> tri à bulle » qui vérifie l'hypothèse de recherche.

Pour conclure, la méthode de tri la plus adaptée sur la liste chainée est le tri par fusion par son efficacité. Mais au vu de sa complexité à être mis en place, si nous comptons travailler sur des listes déjà trier en décroissant, nous pouvons le remplacé par le tri par insertion. Et si nous souhaitons travailler sur des listes quelconques ou croissantes, nous pouvons l'échangé par le tri rapide.