

TP1 - IFT2015

- Analyse -

Tristan Savaria

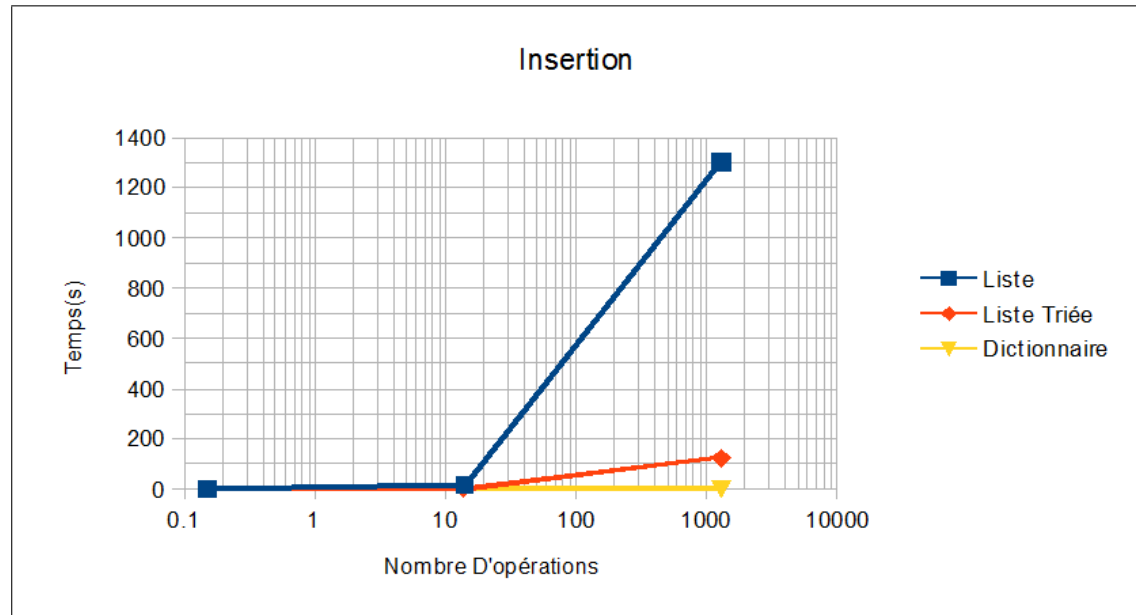
9 février 2014

Table des matières

1	Résultats et Conclusion	3
1.1	Insertion	3
1.2	Recherche	4
1.3	Suppression	5
2	Commentaire	6

1 Résultats et Conclusion

1.1 Insertion

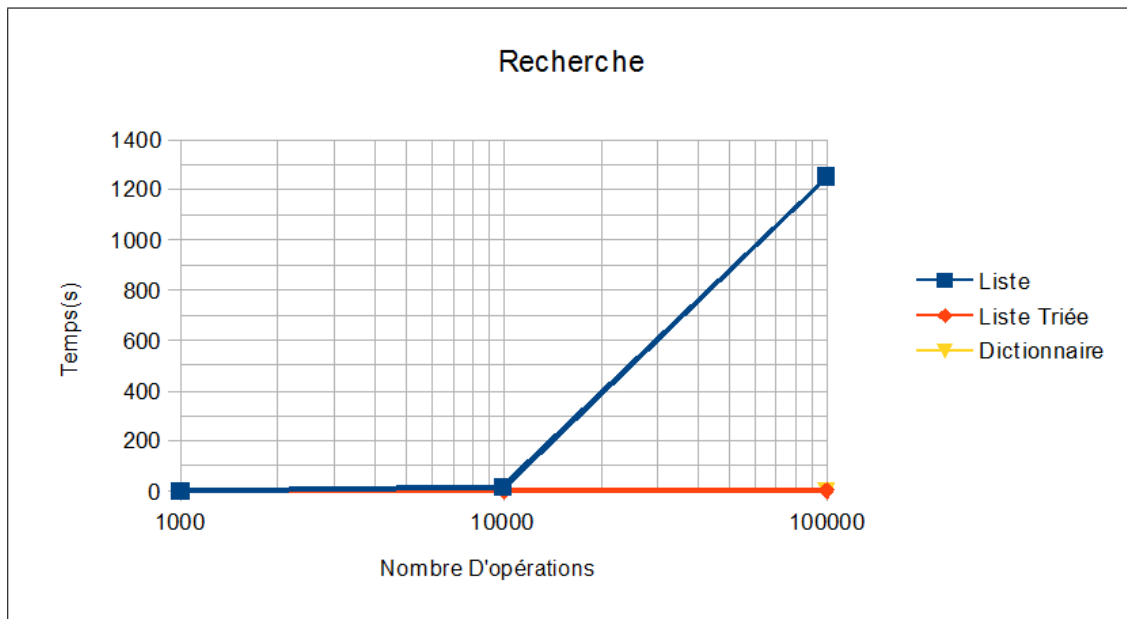


On observe que l'insertion dans la structure de Liste consomme beaucoup plus de temps que les deux autres structures. Une liste ajoute habituellement un élément dans l'ordre de $O(1)$. Cependant, dans le cadre de ce projet, chaque insertion requiert une recherche pour trouver si le Mot y est déjà présent pour incrémenter son compte si c'est le cas. La recherche pour la liste est dans l'ordre de $O(n)$ et c'est pour cela que l'insertion dans la structure de Liste est aussi lente.

La liste triée arrive seconde puisqu'à chaque insertion une recherche binaire est effectuée pour insérer/trouver l'élément au bon index.

1.2 Recherche

La recherche dans la structure liste est la plus lente, car au pire cas nous devons comparer chaque élément de la liste à celui que nous cherchons ce qui est dans l'ordre $O(n)$. Liste Triée et Dictionnaire sont confondues par une erreur de formatage de ma part, cependant Dictionnaire est plus rapide avec 0.04 sec pour 100 000 opérations contre 2.64 sec pour la Liste Triée. Liste Triée utilise une recherche binaire, cependant le hashtable de dictionnaire est plus rapide.



1.3 Suppression

La suppression est plus lente pour la Liste. Comme pour l'insertion, une recherche est effectuée sur la structure pour trouver l'élément à supprimer. Liste Triée et Dictionnaire sont confondues par une erreur de formatage de ma part, cependant Dictionnaire est plus rapide avec 0.07 sec pour 100 000 opérations contre 2.54 sec pour la Liste. Nous pouvons observer que la suppression a été aussi rapide que la recherche pour la Liste Triée et le dictionnaire.

2 Commentaire

Il est possible que le code que j'ai utilisé pour la prise de mesure a eu un impact sur la validité des données. Par exemple, le code utilisé ne générerait pas le cas où l'élément à rechercher et à supprimer n'existait pas dans la structure. La structure de liste est capable de mieux performer dans le cas de l'insertion s'il était possible de ne pas faire de recherche avant l'insertion. Cependant, par la nature de ce travail il est impossible de contourner cette restriction. Avec plus d'éléments il serait possible d'observer le cas où la Liste Triée et le Dictionnaire ne sont plus confondue en une seule ligne.

