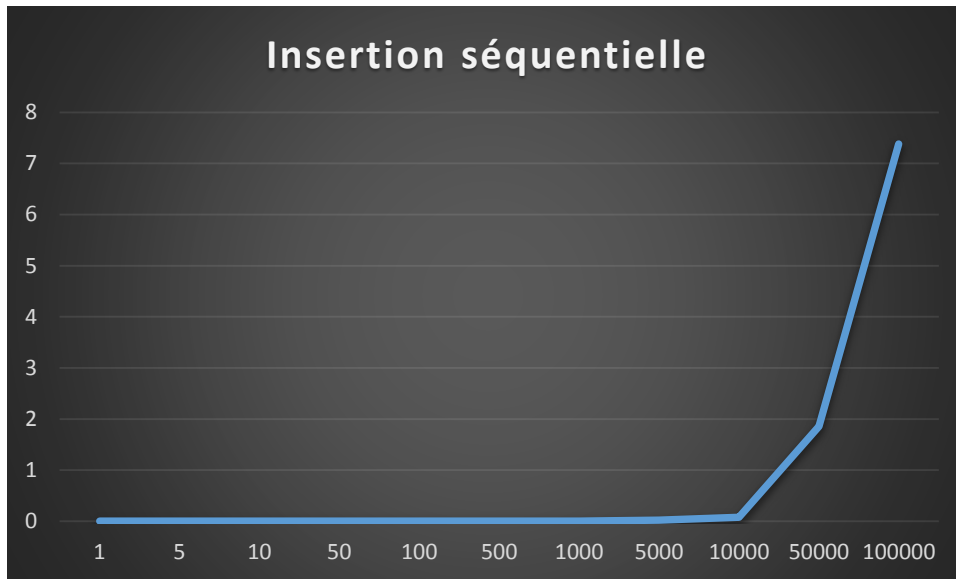


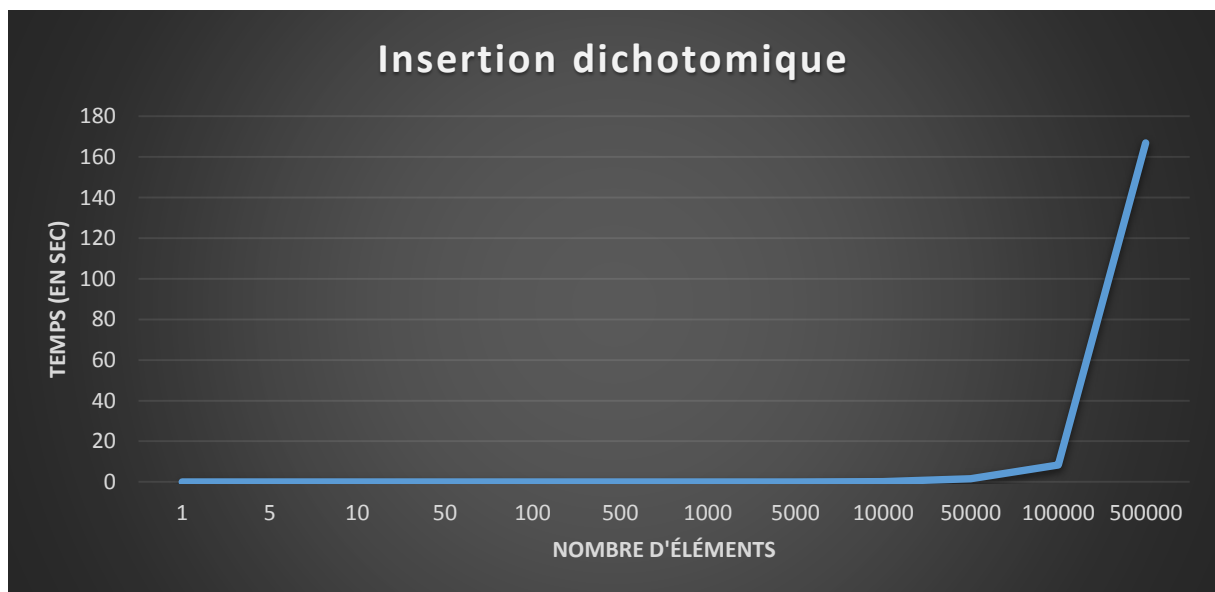
Insertion séquentielle

Voici les résultats obtenue sur un tableau ayant des valeurs de 0 à 100 celons le temps en secondes, avec l'algorithme vu en cours. On passe le cap des 5min d'exécution par tri pour 100000 éléments.



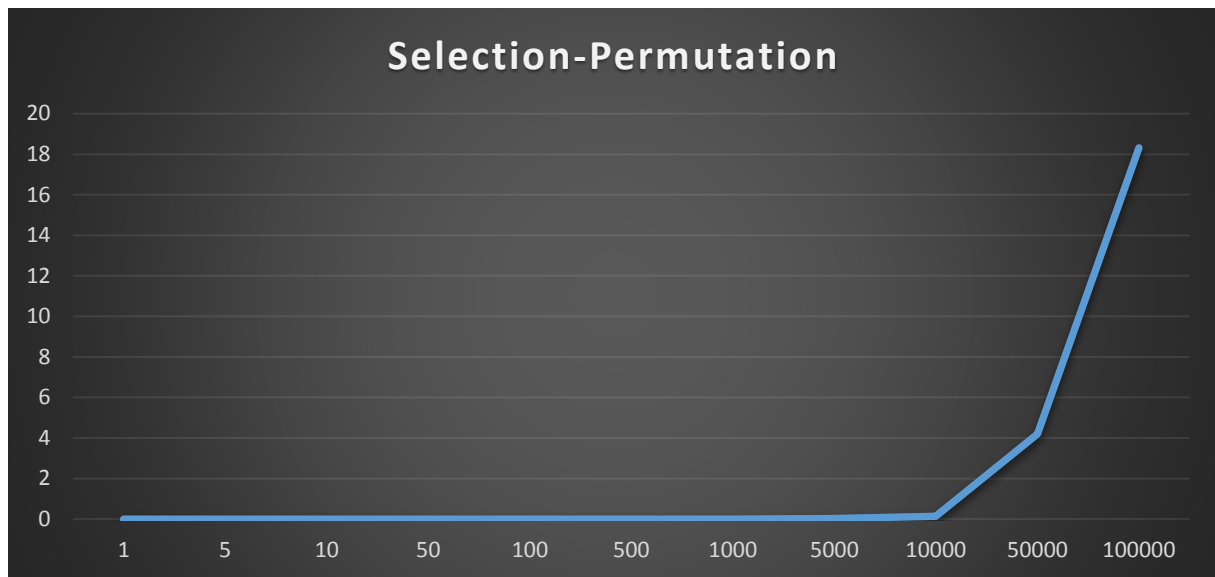
Insertion dichotomique

Voici les résultats obtenue sur un tableau ayant des valeurs de 0 à 100 celons le temps en secondes, avec l'algorithme vu en cours. Pour ce tri, nous passons le cap des 5min pour 1000000 éléments.



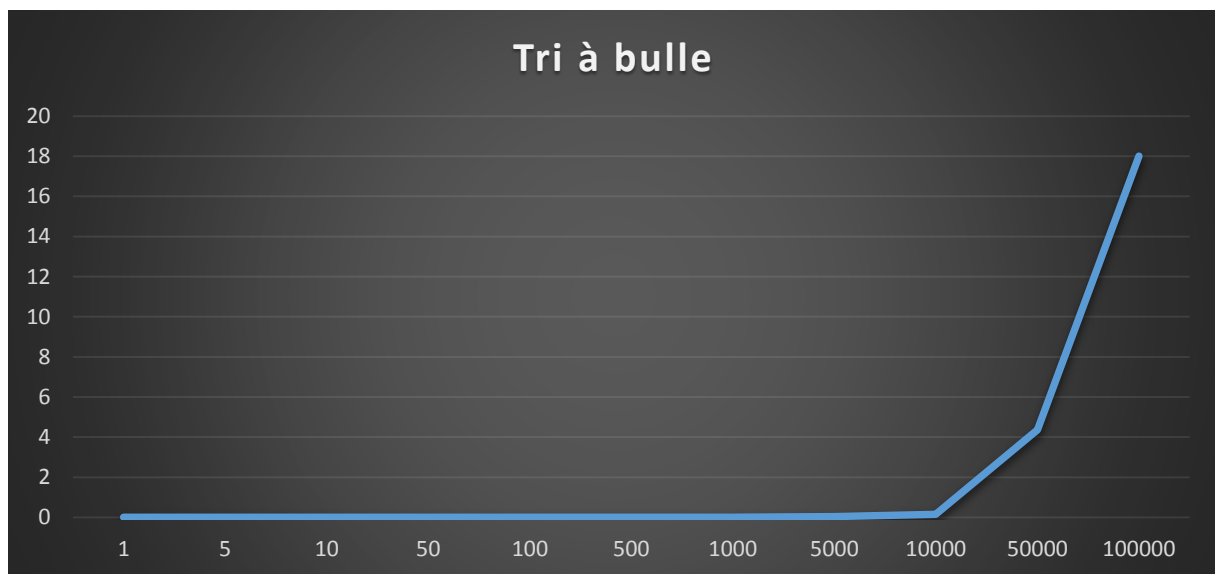
Tri par sélection permutation

Voici le résultat obtenu avec ce tri pour un tableau de 1000000 éléments.



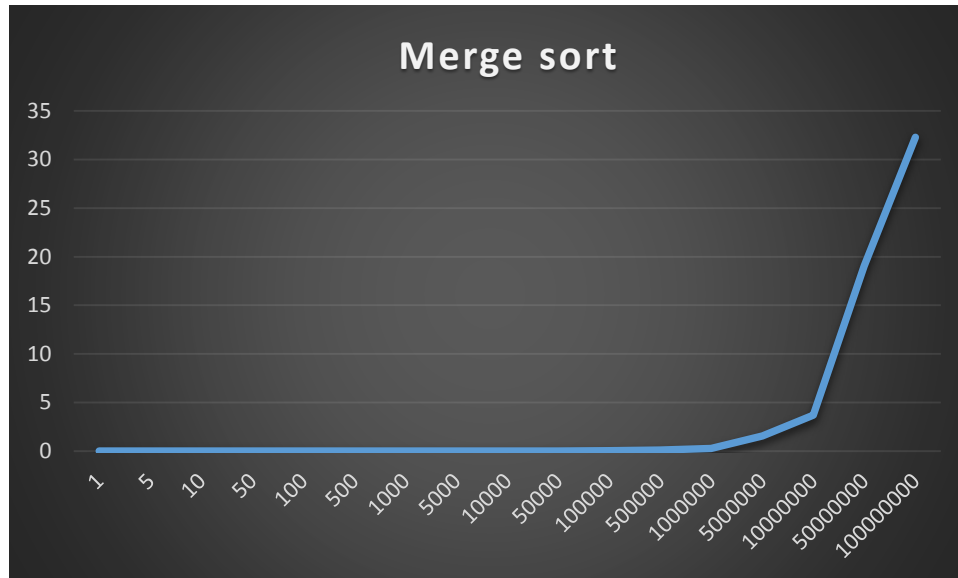
Tri à bulles

Voici les résultats obtenus pour le tri à bulle, nous atteignons les 5min d'exécution à partir de 1000000 éléments :



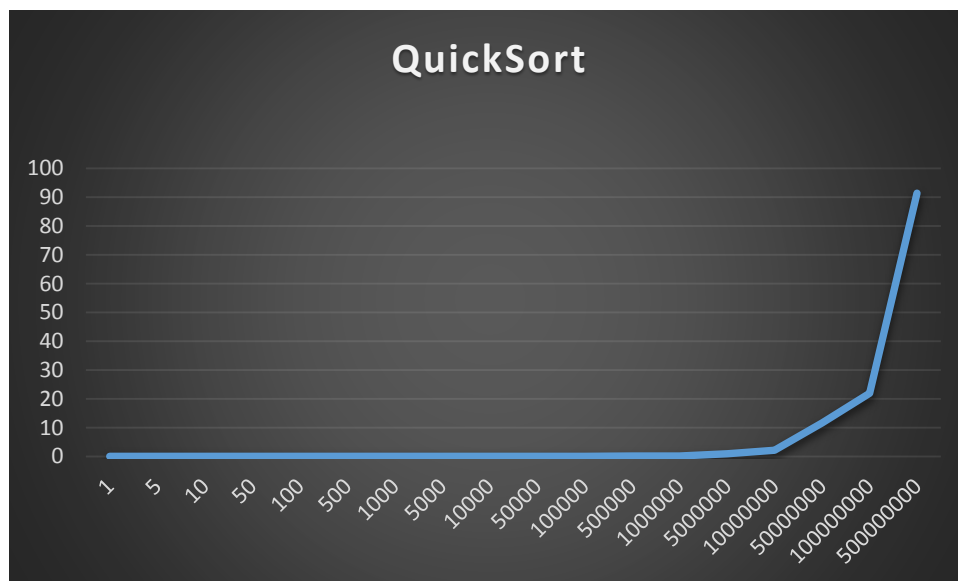
Tri par fusion

Dans la gamme des tris rapide nous avons le tri par fusion vu en TD, voici les résultats obtenu pour un tableau ayant des valeurs de 0 à 100, je ne suis pas allé assez haut en nombre d'élément pour atteindre les 5min d'exécution.



Tri rapide

Pour ce tri, de même que le tri fusion nous n'avons pas pu attendre les limites de 5min pour ce tri.



Le tri par base

Ce tri consiste de regarder à trier des valeurs celons le nombre de symbole d'un chiffre courant. Visuellement cela donne :

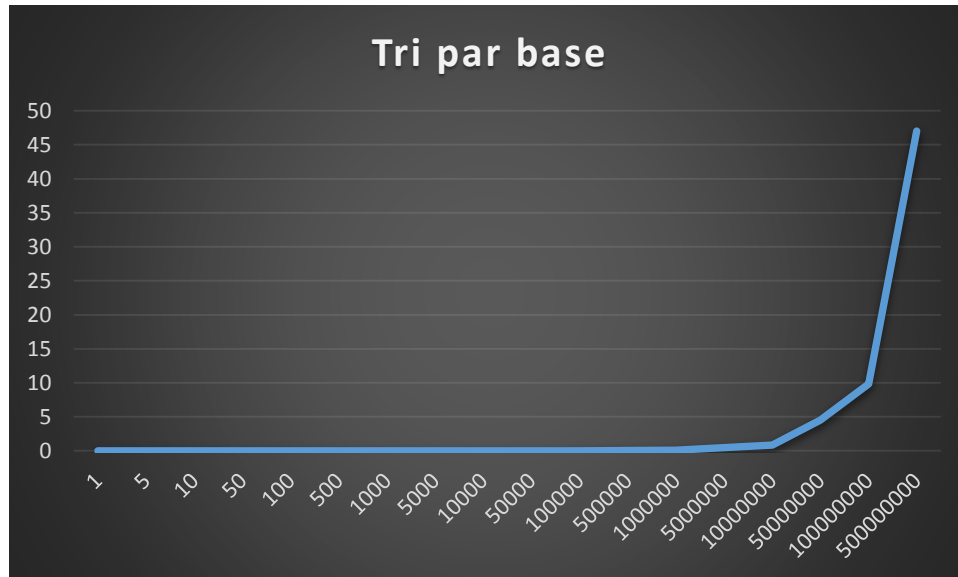
Première étape :

25	3	15	23
3	25	VIDE	VIDE
23	15	VIDE	VIDE

Deuxième étape :

3	23	25	15
3	15	23	25

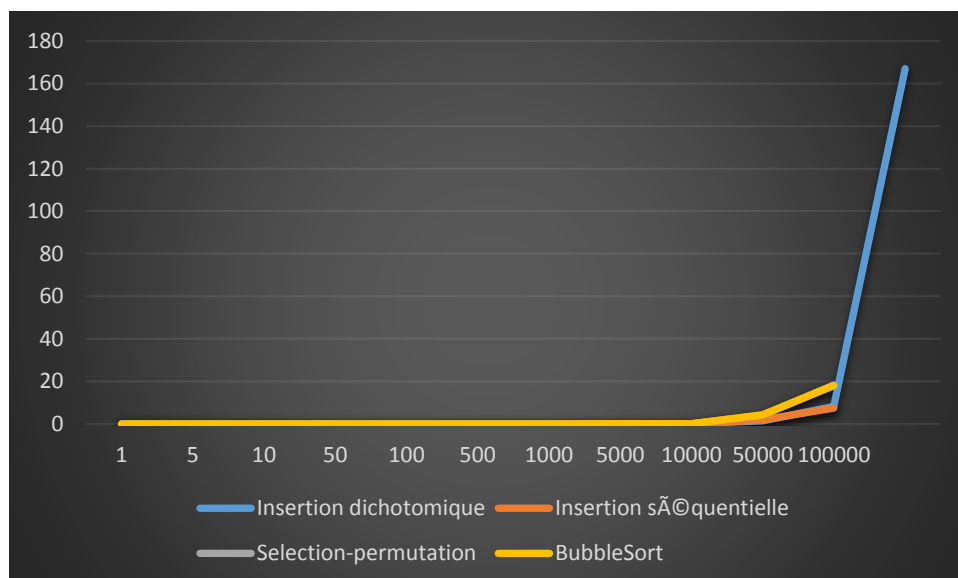
Voici les résultats obtenu lors de mes tests :



On peut en déduire que ça complexité est de : $\in \theta(n \log n)$

Comparatif tri lent

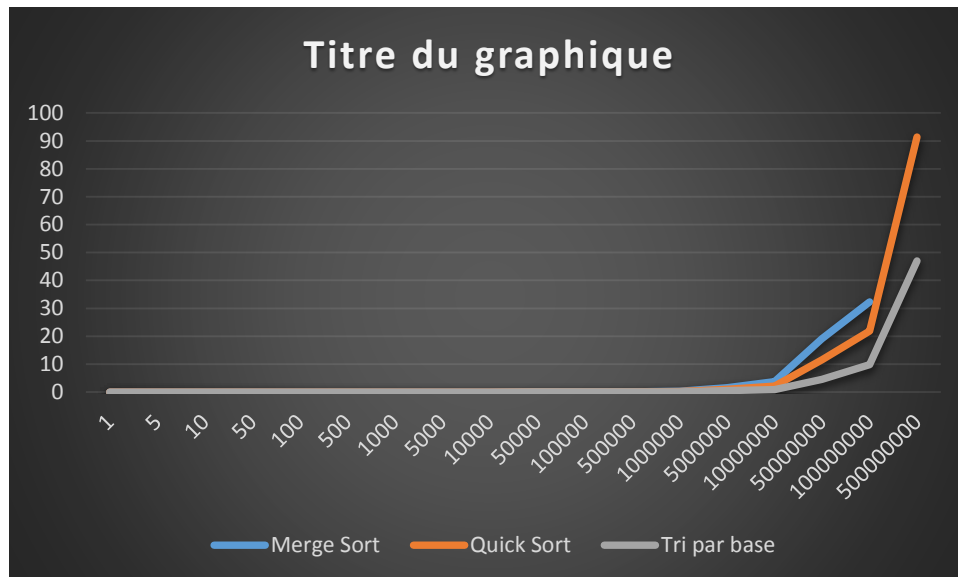
Voici un graphique contenu un comparatif parmi les tris les plus lents :



On en déduit que parmi ces tris l'insertion dichotomique est plus performant avec un tableau ayant des valeurs au maximum de 100 et au minimum de 0. Le moins performant est le tri à bulle car celui-ci passe le cap des 5min plus rapidement.

Comparatif tri rapide

Pour le même tableau que pour les tris lents on se rend compte que le tri le plus performant est le tri par base du au faite que les valeurs du tableau n'allant pas plus de 100. On retrouve ensuite le tri rapide en termes de rapidité de calcul.



Je pense que le tri le plus stable reste le tri rapide, car le tri par base dépendra des valeurs des cases dans le tableau

Conclusion

Ce projet m'a permis de découvrir et de redécouvrir différents algorithmes de tri plus ou moins performants. Savoir la rapidité d'un algorithme de tri peut-être très important dans le domaine professionnel. L'implémentation de certains algorithmes peut être délicate et peut être une perte de temps si le temps d'exécution n'est pas important pour une application. On se rend compte aussi que les tris les plus performants sont les plus complexes à implémenter.