**Mesures de temps d’exécution d’algorithmes de tri et de recherche avec Python**

*BONNAIRE Lucas & RENAULT Rémi*

*16/12/2024*

Toutes les mesures ont été prises sur la base de scripts Python exécutés sur un MacBook Pro M4 Pro avec 24Go de RAM.

Le code source utilisé pour ces mesures est disponible sur ce repository GitHub :

<https://github.com/CORT1N/esgi-algorithmic-works>

Toutes les mesures ont été moyennées sur la base de 100 exécutions.

Table des matières

[1. Le tri à bulles 3](#_Toc185280091)

[2. Le tri par sélection 4](#_Toc185280092)

[3. Le tri par insertion 5](#_Toc185280093)

[4. Le tri rapide 6](#_Toc185280094)

[5. Le tri hollandais 7](#_Toc185280095)

[6. Le tri fusionné 8](#_Toc185280096)

[Interprétation des résultats pour les algorithmes de tri 9](#_Toc185280097)

[7. La recherche dichotomique 10](#_Toc185280098)

[8. La recherche linéaire 11](#_Toc185280099)

[Interprétation des résultats pour les algorithmes de recherches 12](#_Toc185280100)

# Le tri à bulles

Le tri à bulles compare et échange les éléments adjacents d’une liste jusqu’à ce que celle-ci soit triée.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Le tri par sélection

Le tri par sélection consiste à trouver l’élément le plus petit (ou le plus grand) à chaque étape et à le placer à sa position correcte dans la liste.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Le tri par insertion

Le tri par insertion consiste à insérer chaque élément dans sa position correcte par rapport aux éléments déjà triés, en les déplaçant progressivement.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Le tri rapide

Le tri rapide consiste à choisir un pivot, partitionner la liste autour de ce pivot, puis trier récursivement les sous-listes à gauche et à droite du pivot.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Le tri hollandais

Le tri hollandais (ou tri de Dijkstra) est un algorithme qui partitionne une liste en trois groupes distincts (par exemple, inférieur, égal et supérieur à un pivot) afin de trier efficacement des données avec des valeurs répétées ou limitées.

Par son fonctionnement, il est limité à 3 valeurs distinctes maximum, ce qui oblige les mesures à se restreindre à l’intervalle [0, 2].

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



# Le tri fusionné

Le tri fusionné divise récursivement la liste en sous-listes, puis fusionne les sous-listes triées pour obtenir une liste finale triée.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Interprétation des résultats pour les algorithmes de tri

Pour le tri à bulles, on remarque surtout qu’avec 1000 valeurs, quelle que soit l’intervalle de valeurs utilisée, il devient beaucoup plus lent.

Même affirmation pour le tri par sélection et le tri par insertion qui divisent cependant pour la même quantité et la même intervalle le temps d’exécution par deux.

Concernant le tri rapide, il porte bien son nom, mais surtout devient de plus en plus efficace à mesure que l’intervalle de valeurs utilisée s’agrandit.

Enfin, pour le tri fusionné, on remarque qu’il est nettement meilleur et plus stable que tous les autres, sans dépasser une seule fois 1ms de temps d’exécution.

Nous n’avons pas abordé le tri hollandais qui n’est pas une mesure cohérente comparée aux autres, ne s’appliquant qu’aux intervalles très petites d’un maximum de trois.

Nous pouvons conclure de nos observations que le tri fusionné est le plus efficace en toutes circonstances selon ces mesures.

Le tri rapide le concurrence très facilement à mesure que l’intervalle de valeurs utilisées s’agrandit.

Les autres types de tri sont plus efficaces pour de petites quantités de données.

# La recherche dichotomique

La recherche dichotomique (ou binaire) consiste à diviser récursivement une liste triée en deux moitiés pour localiser un élément en réduisant à chaque fois l’espace de recherche.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# La recherche linéaire

La recherche linéaire consiste à parcourir séquentiellement chaque élément d’une liste jusqu’à trouver l’élément recherché.

Pour 10 valeurs entre 0 et 2 :



Pour 100 valeurs entre 0et 2 :



Pour 1000 valeurs entre 0 et 2 :



Pour 10 valeurs entre 0 et 9 :



Pour 100 valeurs entre 0 et 9 :



Pour 1000 valeurs entre 0 et 9 :



Pour 10 valeurs entre 0 et 99 :



Pour 100 valeurs entre 0 et 99 :



Pour 1000 valeurs entre 0 et 99 :



Pour 10 valeurs entre 0 et 999 :



Pour 100 valeurs entre 0 et 999 :



Pour 1000 valeurs entre 0 et 999 :



# Interprétation des résultats pour les algorithmes de recherches

Contrairement aux algorithmes de tri où les écarts étaient flagrants avec les mêmes intervalles et quantités de valeurs, ici les algorithmes de recherche s’avèrent efficaces en toutes circonstances.  
  
Un point intéressant cependant à noter est que l’algorithme de recherche linéaire s’avère légèrement moins puissant lorsqu’on atteint 1000 valeurs.

Même après avoir testé avec beaucoup plus de valeurs, où une intervalle plus grande, les deux algorithmes ne perdaient pas en efficacité.