



# Benchmarking d'algorithmes de tris :

- ✓ tri insertion
- ✓ tri sélection
- ✓ tri bulles
- ✓ tri fusion
- ✓ tri rapide

DIU, Bloc 2 : Projet

## Activité 3, question 1

Groupe :

Delrot Sylvain

Houunkpatin Bonaventure

Jaubert Philippe

## 1 But

Étudier la complexité en temps et en place des algorithmes de tris en les exécutant sur des tableaux de tailles différentes.

Les coûts des différentes exécutions des différents tris seront stockés et affichés de façon graphique.

### Livrables :

- un programme de benchmarking écrit en python (tp\_benchmark\_nb\_operation\_tout.py);
- les résultats des analyses (*Comparaison\_méthode\_tri.pdf*) ;
- le présent fichier d'analyse des résultats (*Projet3\_Question1.pdf*).

## 2 Analyse des résultats

Grâce à l'analyse des temps de tri en fonction de la taille des listes mélangées créées, on peut déjà proposer un premier classement des méthodes étudiées (temps de tri de 2000 nombres sur pentium core i3):

Moins performant	Plus performant
tri bulle (~0.5s) < insertion ou sélection (~0,2s)	tri rapide et fusion (t<0.015s)

### 1. Tri sélection et tri insertion

- a) Les méthodes de tri par sélection et insertion semblent à peu près équivalentes du point de vue temps de tri.
- b) Elles n'ont pas la même sensibilité au "dérangement" des données :
  - *Que les données soient un peu ou complètement dérangées, le tri sélection ne change pas notablement le temps de tri : il va toujours piocher la valeur minimale dans ce qui reste à trier en parcourant toutes les valeurs non triées, cela se voit aussi sur le nombre d'opérations.*

## 2. Tri fusion - tri bulle – tri rapide

- a) Les résultats pour le tri fusion sont intéressants :
- soit il est très performant comparé aux autres méthodes étudiées (temps "quasi nul" à cause de l'échelle sur la courbe)
  - soit il est très mauvais car le principe du tri fusion est de diviser les listes jusqu'au bout pour aboutir au comparaison 2 à 2.
- b) En termes de nombre d'opérations, le tri fusion fait grosso modo 50 fois mieux que le tri bulle pour des listes de 2000 nombres. Il semble être légèrement plus performant que le tri rapide (les nombres constatés sont du même ordre de grandeur)
- c) En traçant le logarithme du nombre d'opérations en fonction du logarithme de la taille des données, on peut montrer que les performances du tri fusion dépendent linéairement de la taille des données alors que celles des autres méthodes étudiées sont polynomiales ce qui rend ces méthodes rapidement inutilisables pour un trop grand nombre de données.

### **Remarques :**

*Réalisable au niveau terminale (une fois la fonction logarithme étudiée en cours de mathématiques) :*

- ➔ *produire le graphique  $\log(t)=f(\log(\text{taille}))$*
- ➔ *produire  $\log(\text{opération})=f(\log(\text{taille}))$*
- ➔ *et effectuer une régression linéaire pour l'exposant de la loi de puissance qui correspond à la complexité (polynomiale?) des différentes méthodes.*

### **Conclusions pratiques :**

On peut utiliser les résultats pour imaginer le choix à faire entre plusieurs méthodes de tri parmi les 4 étudiées. Le tri fusion serait la méthode la plus efficace.