

# SAE 1.02 - Comparaison Algorithmique

Projet n°2  
Équipe n°21  
Rendu n°1

Justine BONDU  
Jules CHUZEVILLE  
Alizée LEBARON

---

Dans un premier temps, pour tester les dits programmes dans les conditions les plus optimales selon nous :

- Éteindre toutes applications parasites allumées sur l'ordinateur.
- Mettre l'ordinateur en mode avion
- Vérifier les tâches en fond avec le gestionnaire de tâche et éteindre celles inutiles au fonctionnement de la machine.

Ces précautions certes précises nous permettent de limiter les incohérences due à l'utilisation du processeur par d'autres applications tournant en fond pendant le test du code.

Choix de la donnée dans les tableau pour les tries:

- Utiliser les **booleans** aurait simplement été de trier un tableau sur deux valeur possible ce qui n'est pas très parlant.
- Utiliser les **string** aurait été assez long à trier surtout que ça n'aurait à notre goût pas beaucoup de sens (peut-être trier en prenant en compte la longueur ?).
- Utiliser des **char** aurait peut-être été parlant, toutefois, nous les aurions comparé au code ascii, et, ce n'est pas un ordre qui nous est familier. Surtout en prenant en compte par exemple que les majuscules sont plus petites que les minuscules.
- Nous choisissons donc d'utiliser les **entiers** afin de faciliter la lecture et la compréhension du tri par le programme. Pour le cerveau humain, c'est plus simple de comparer

Maintenant que nous avons le type de variables, il nous faut déterminer parmi les quatres possibilités : Byte, Short, Int, Long.

Nous prendrons le type int, car nous avons pu tester précédemment son efficacité par rapport aux autres types d'entiers (environ 23× plus rapide tout de même).

Vérification que les tri ont fonctionné:

On crée deux tableaux entiers de très grande taille (50 entiers différents), le premier est en désordre, le deuxième est trié. On envoie le tableau non rangé dans la méthode tri. Il renvoie le tableau trié. Le programme principal vérifie le tableau nouvellement rangé avec celui déjà rangé.

Pour vérifier que les tris fonctionnent, nous utiliserons à chaque fois les deux mêmes tableaux : Un déjà trié et un en désordre.

Le premier nous permettra de vérifier que le programme ne détruit pas les données et le deuxième de confirmer qu'il l'est trié.

Fonctionnalité supplémentaire de la class Test :

La classe test calculera aussi le temps entre l'envoi de la réponse dans le sous-programme et le retour de réponse.

Le temps en question sera calculé en nanosecondes pour plus de précision en termes de données.