**Exercice 1 : tri et test**

Par Lucas GAIO DOS SANTOS  
Master 1 Data Science  
2022-2023

# Enoncé

1. Implémentez un algorithme de tri par sélection en langage Java (ou python)

2. Implémentez la méthode de test cet algorithme

3. Pour la méthode de test, testez les cas où le tableau est vide, où les éléments du tableau sont en désordre et où ils sont en ordres.

4. Vérifier la couverture du code de votre méthode et faites des captures d’écran que vous ajouterez au document final qui me sera envoyé.

5. Implémentez une méthode main qui prendre en input des chiffres saisies au clavier et qui affichera le résultat du tri.

# Tri par sélection

Fait en python, j’ai décidé à l’origine de faire avec la méthode « sorted() », je propose ensuite la méthode « sort() » et fait à la main.

Une image contenant texte

Description générée automatiquement

# Implémentation du test

J’ai décidé de faire deux tests en un, un premier où la sortie est parfaitement égale, et la seconde où il y a une erreur. Comme nous le verrons par la suite, le test est valide.

Une image contenant texte

Description générée automatiquement

# Test vide, en désordre et en ordre

Le test de base est comparable au test en désordre, je ne souhaite pas faire doublon (voir ci-dessus). Voici donc les deux autres :

Une image contenant texte

Description générée automatiquement

# Couverture du code

Dans cette première photo, nous voyons que tout le code de test a bien été lancé, un total succès.

Une image contenant texte

Description générée automatiquement

Dans cette deuxième photo, vous avez tout mon code, la partie demandé dans les énoncés précédents ont été lancés avec succès (voir « trie\_croissant() »).

Une image contenant texte

Description générée automatiquement

Dans la capture d’écran suivante, c’est un récapitulatif de ce qui a été lancer, 47% pour le main et 100% pour test\_trier, ce qui fait une moyenne de 65%.

Une image contenant texte, noir, écran, capture d’écran

Description générée automatiquement

# Méthode pour capter la saisie et faire le tri.

Pour capter la saisie de l’utilisateur, j’ai utilisé la méthode suivante :

Une image contenant texte

Description générée automatiquement

J’utilise aussi un sélecteur dans le CMD, que j’ai fait comme ceci :

Une image contenant texte

Description générée automatiquement

Ces deux méthodes sont appelées par le constructeur que voici :

Une image contenant texte

Description générée automatiquement

Suite à l’initialisation, nous appelons les méthodes adaptées.

# A améliorer

J’aimerais faire des tests pour les méthodes avec « Self », ce que je ne sais pas encore faire, je me renseigne en ce moment.

Je vais sortir un exécutable pour rendre plus utilisable le code et donc rendre accessible les méthodes sans ligne de code.

Je vais chercher à optimiser et améliorer le tout.

J’ai des corrections de faute d’orthographe à faire.

# Retrouver le code

Voici le lien de dépôt : <https://github.com/Lucas-formation/python.git> (pour retrouver le programme adapté, allez dans « programme basique », puis « trier », tout le code est dedans.). Les autres fichiers sont d’autres programmes, et tests fait sur python. C’est mon dépôt de cours et d’expérimentation.

Pour retrouver la couverture, allez dans « htmlcov » et lancez les fichiers HTML.

Ce document word est aussi accessible sur le dépôt, si je parviens à faire la vidéo, elle serait elle aussi disponible au même endroit.