

# Algorithmie et structures de données

## TD2 : Algorithmes de tri

---

### Objectifs du TD

Implémenter les algos de tri vus en cours.

Comparer le temps de calcul d'un programme.

Maîtriser malloc et free.

---

### Consignes

Pas d'IDE. Utilisez un logiciel de texte simple avec coloration syntaxique. Il y en a pour tous les OS (gedit pour Linux, Notepad++ pour Windows, CodiEditor pour MacOS par exemple).

Pas de compilation compliquée, un simple gcc!

Pas d'appel à l'IA ou de copier/coller d'Internet. Ce que nous faisons est la base, vous trouveriez la solution directement sans comprendre et donc sans apprendre ...

Lisez bien l'énoncé, il contient beaucoup d'informations et chaque mot est utile!

Aidez-vous de la documentation lorsque je vous en indiquerai. Lorsqu'il utilise une nouvelle librairie, un développeur passe beaucoup de temps dans la documentation pour comprendre son fonctionnement. RTFM!

Coder proprement avec des noms de variables et de fonctions bien choisis. Ajoutez des commentaires utiles à votre code afin de pouvoir encore le comprendre dans quelques jours.

Respecter les coding style. Je vous conseille de suivre les conseils de la fondation GNU<sup>1</sup> ou de CS50<sup>2</sup>.

Simple is beautiful. N'essayez pas de coder compliqué, restez simple, cela sera plus efficace et moins sujet aux erreurs.

Réfléchissez avant de coder, prenez un peu de temps pour dessiner/écrire vos idées sur une feuille. Le temps de la réflexion que vous prendrez, vous économisera énormément de temps de debug.

---

## 1 Préparation

1. Récupérer le squelette sur Moodle et lire le code
2. Afin que la suite puisse fonctionner vous devez compléter les fonctions suivantes présentes dans `utils.c` :
  - `copy_table` qui copie la table donnée en argument
  - `verification_tri` qui vérifie si la table en argument est triée ou non

---

1. GNU : [https://www.gnu.org/prep/standards/html\\_node/Writing-C.html](https://www.gnu.org/prep/standards/html_node/Writing-C.html)

2. CS50 : <https://cs50.readthedocs.io/style/c/>

## 2 Algorithmes de tris

1. Implémenter l'algorithme de tri par sélection
2. Implémenter l'algorithme de tri par insertion
3. Implémenter l'algorithme de tri à bulles
4. Implémenter l'algorithme de tri fusion
5. Implémenter l'algorithme de tri rapide
6. Implémenter l'algorithme de tri par comptage
7. Implémenter l'algorithme de tri par base
8. Implémenter l'algorithme de tri par paquets

## 3 Limitation du nombre d'éléments

Vous avez fini? Bravo! Je vous donne un exercice bonus.

1. Quelle est la taille maximum d'éléments ( $n$ ) que l'on peut trier avec ce programme? Pourquoi?
2. Comment pourrait-on résoudre ce problème?
3. Modifiez le programme pour appliquer votre idée.