**Semaine tri**

Matière

* Les algorithmes de tri

Objectifs

* Implémenter les algorithmes de tri vus au cours : tri par comptage et tri par sélection
* Pouvoir les reproduire tels quels ou des variantes de ceux-ci

Mise en garde

La tentation est grande de prendre de recopier tel quel un algorithme trouvé dans le syllabus ou sur le net. Ce n’est pas une bonne idée !

De même, pour l’examen n’apprenez pas par cœur les algorithmes vus au cours.

Vous devriez, plutôt que de recopier ou mémoriser du *code*, **comprendre** et retenir les ***principes moteurs*** de l’algorithme et vous **entraîner** à reconstituer celui-ci.

Conseil

Assurez-vous d’avoir bien compris les principes moteurs des algorithmes demandés.

Allez sur le site : <https://visualgo.net/en/sorting>.

## Exercices obligatoires

A Tri par comptage

Complétez la méthode tri(int[], int borneSup) de la classe *TriParComptage*.

La méthode main de cette classe permet de tester la méthode demandée.

B Tri par sélection

Complétez les méthodes triSelf (int[] t) , int[] tri (int[] t) et tri5() de la classe *TriParSelection.*

La méthode main de cette classe permet de tester les méthodes demandées.

# Exercices supplémentaires

A Complétez les méthodes triSelf(int[] t, int borneInf, int borneSup) et

triSelf(int[] t)) de la classe *TriParComptage*.

B Complétez la méthode triPartiel(int[] t, int indiceDebut, int indiceFin) de la classe *TriSelection*.

# Exercice « défi »



B2 Implémentez la méthode tri5() de la classe *TriParSelection* en suivant le défi :

La table passée en paramètre n’est pas modifiée **et** aucune table intermédiaire n’est utilisée à l’exception de celle renvoyée.

**Astuce :** pensez à retenir l’indice du dernier minimum inséré pour gérer le problème des ex-aequo.