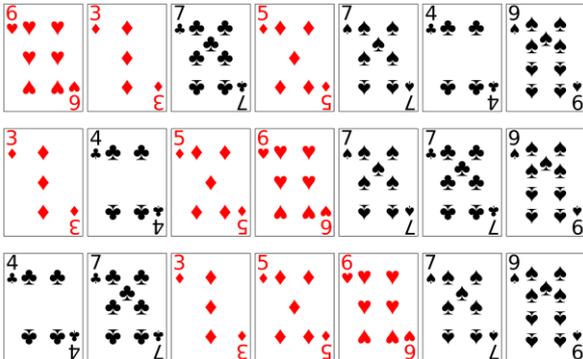


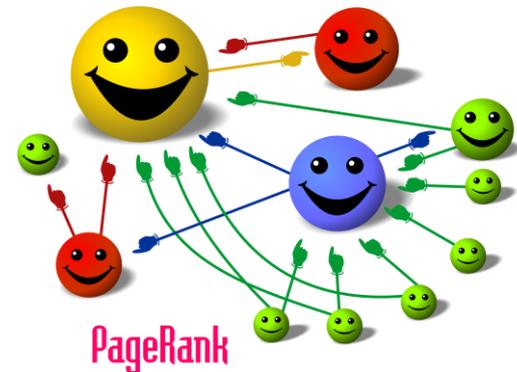
## Bloc 2 - Algorithmique

# Les algorithmes de tri

## De la sélection...



## ... au Quicksort



- 1. Introduction**
- 2. Objectifs pédagogiques de la séquence**
- 3. Séquence proposée**
- 4. Liens ressources et références**

# 1. Introduction

- **Organisation matérielle**
  - équipement informatique relié à Internet pour chaque élève ;
  - des jeux de 52 cartes (1 pour 4 élèves suffit).
  
- **Logiciels installés**
  - EDI pour Python ;
  - Navigateur, suite bureautique.
  
- **Bibliothèques Python**
  - *Matplotlib* (présentation graphique) ;
  - *Random* ;
  - *Math*.

## 2. Prérequis - Chronologie - Objectifs de la séquence

- **Prérequis : séquence(s) autour des points suivants :**
  - Représentation des données – types et valeurs de base ;
  - Maîtrise d'un EDI Python (Thonny, Spyder...) ;
  - Maîtrise suffisante des concepts d'affectation, de fonctions et d'itérations en Python ;
  - Opérateurs booléens ;
  - Langages de programmation – constructions élémentaires.
  
- **Chronologie**
  - Idéalement au 3<sup>ème</sup> trimestre.

## 2. Objectifs de la séquence - suite

### ■ Objectifs

- Connaissances
  - *De la nécessité de trier les données ;*
  - *Définir* ce qu'est un algorithme ( y compris approche historique).
- Compétences
  - *Comprendre* une problématique ;
  - *Découvrir* les notions grâce à des activités débranchées (cartes à jouer) – activité en binôme ;
  - *Découvrir* et *exploiter* un code existant mettant en œuvre un algorithme de tri « simple » (sélection) ;
  - *Implémenter* et *exploiter* un algorithme de tri « simple » (insertion) en Python – Analogie avec un code existant ;
  - *Mesurer* et *comparer* des temps d'exécution.

## 2. Objectifs de la séquence - suite

: liens du B.O. avec la séquence

### ■ Références avec le programme de NSI

Cet enseignement a vocation à multiplier les occasions de mise en activité des élèves, **sous des formes variées** qui permettent de développer des compétences transversales :

- faire preuve d'autonomie, d'initiative et de créativité ;
- présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat ;
- coopérer au sein d'une équipe dans le cadre d'un projet ;
- rechercher de l'information, partager des ressources ;
- faire un usage responsable et critique de l'informatique.

Tri d'une table

Trier une table suivant une colonne.

Une fonction de tri intégrée au système ou à une bibliothèque peut être utilisée.

### Algorithmique

Le concept de méthode algorithmique est introduit ; de nouveaux exemples seront vus en terminale. Quelques algorithmes classiques sont étudiés. L'étude de leurs coûts respectifs prend tout son sens dans le cas de données nombreuses, qui peuvent être préférentiellement des données ouvertes.

Il est nécessaire de montrer l'intérêt de prouver la correction d'un algorithme pour lequel on dispose d'une spécification précise, notamment en mobilisant la notion d'invariant sur des exemples simples. La nécessité de prouver la terminaison d'un programme est mise en évidence dès qu'on utilise une boucle non bornée (ou, en terminale, des fonctions récursives) grâce à la mobilisation de la notion de variant sur des exemples simples.

## 2. Objectifs de la séquence - suite

### ■ Références avec le programme de NSI - suite

Tris par insertion, par  
sélection

Écrire un algorithme de tri.

Décrire un invariant de boucle  
qui prouve la correction des tris  
par insertion, par sélection.

La terminaison de ces algorithmes  
est à justifier.

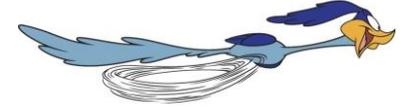
On montre que leur coût est  
quadratique dans le pire cas.

## 3. Séquence proposée

- 1 élève par poste, travail collaboratif, rendu unitaire
- Séquence composée de 2 séances de 2 heures
- **Activité Starter (2 heures) – Starter algo.pdf**
  - Présentation de l'activité (5-10') – **La problématique**
    - **Moteurs de recherche** -> opérations de tri sur de (très) grandes bases de données ;
    - **Problématique** : « Pourquoi chercher à minimiser le temps de réponse d'un moteur de recherche ? »
  - Rappel sur les algorithmes (5-10')
  - **Activité débranchée – tri autour de jeux de cartes (40-45')**
  - **Activité branchée (50-60')**
    - Vidéos présentant des tris
    - Mettre en œuvre le tri par sélection sur un jeu de données (petite taille)
    - Coder le tri par insertion (pseudo-code fourni)
    - Mettre en œuvre le tri par insertion sur un jeu de données (petite taille)
    - Mettre en œuvre le tri par insertion sur un jeu de données de (très) grande taille

## 3. Séquence proposée - suite

- **Activité TGV – Trier à Grande Vitesse (découverte du Quicksort)**
  - Rappel de la séance précédente (5-10')
  - Vidéos sur le tri Quicksort
  - Discussion et questions/réponses
  - Description de l'algorithme (pseudo-code, algorigramme, langage naturel...)
  - Test de la fonction Quicksort fournie sur les grands jeux de données
  - Discussion et questions/réponses - cours
  - Réponse à la problématique

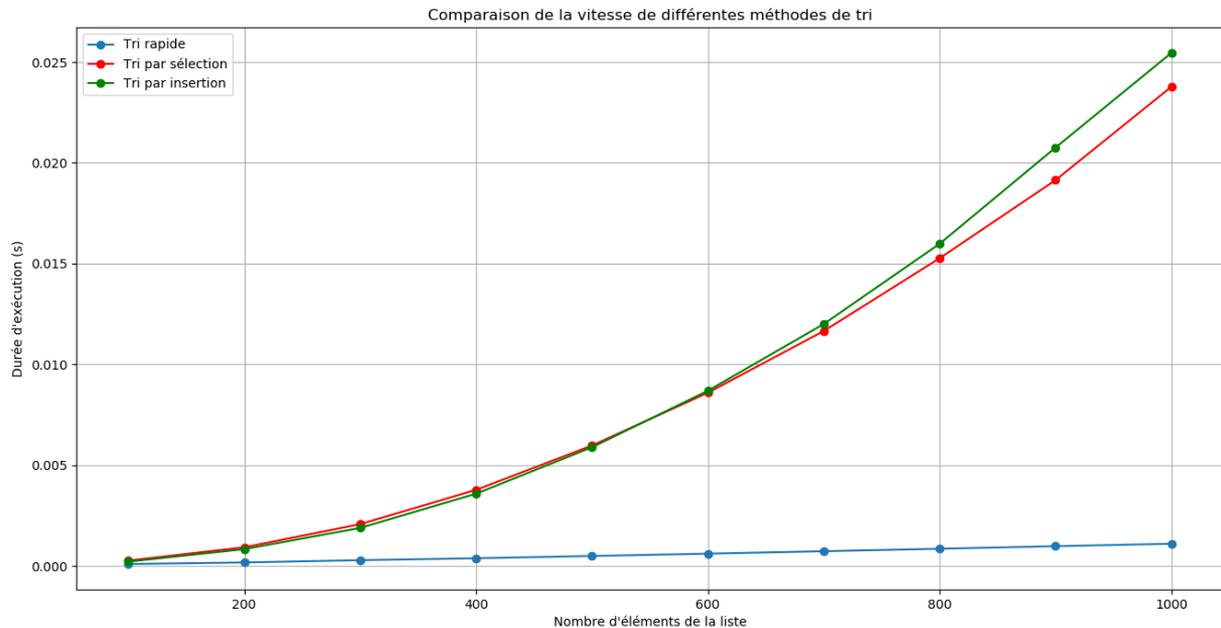


## 3. Séquence proposée - fin

### ■ Réponse à la problématique

« Comment obtenir en temps raisonnable une réponse *triée* d'un moteur de recherche ? »

Les algorithmes naturels (*sélection*, *insertion*) à coût quadratique sont trop lents. Il est nécessaire d'utiliser des algorithmes plus rapides : *Quicksort*, *Fusion*.



## 4. Liens ressources et références

### Liens :

- \* [https://onvaessayer.org/mobileCSP/unit5/5\\_04.php](https://onvaessayer.org/mobileCSP/unit5/5_04.php) - présentations vidéo d'algorithmes de tri ;
- \* [https://fr.wikipedia.org/wiki/Tri\\_fusion](https://fr.wikipedia.org/wiki/Tri_fusion) - le tri fusion ;
- \* <https://interstices.info/les-algorithmes-de-tri/> - les algorithmes de tri pas-à-pas ;
- \* <https://openclassrooms.com/fr/courses/1467201-algorithmique-pour-lapprenti-programmeur/1467940-introduction-au-probleme-du-tri> - introduction aux tris ;
- \* [https://www.youtube.com/watch?v=H98D\\_2qOFho](https://www.youtube.com/watch?v=H98D_2qOFho) – comparaison visuelle des tris (image)
- \* <https://www.youtube.com/watch?v=y9Ecb43qw98> – comparaison visuelle des tris (couleurs)
- \* <https://classic.csunplugged.org/sorting-algorithms/> - activités débranchées complémentaires
- \* <https://www.techiedelight.com/hybrid-quicksort/> - Pour aller plus loin sur le Quicksort (professeur)

### PDF :

- \* Histoire Algo.PDF (Moodle)
- \* Quicksort historical and empirical approaches.PDF (Moodle)

### Livres :

- \* Algorithmique – Cormen, Lierson, Rivest, Stein (Dunod, 2010)