



**HAL**  
open science

## TRIER POUR MIEUX COMPARER

Séverine Fleury, Pascal Lafourcade, Malika More, Mognos Marianne

► **To cite this version:**

Séverine Fleury, Pascal Lafourcade, Malika More, Mognos Marianne. TRIER POUR MIEUX COMPARER. Repères IREM, 2020, 117. hal-04446294

**HAL Id: hal-04446294**

**<https://uca.hal.science/hal-04446294v1>**

Submitted on 8 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# *Repères* **IREM**

**numéro 117**

*Octobre 2019*

*revue*

*des* **Instituts**  
*de* **Recherche**  
*sur l'* **Enseignement**  
*des* **Mathématiques**

*publiée sous le patronage de  
l'Assemblée des Directeurs d'IREM*

**TOPIQUES éditions**

---

## SOMMAIRE

---

Editorial	3
<b><i>Trier pour mieux comparer</i></b>	5
Séverine FLEURY, Pascal LAFOURCADE, Marianne MOGNOS, Malika MORE, Irem de Clermont Ferrand	
<b><i>Rubrique Multimedia</i></b>	26
<b><i>Rubrique Agenda</i></b>	34
<b><i>Aider à la compréhension des fonctions avec un logiciel : les fonctions définies par un algorithme</i></b>	35
Roselyne HALBERT, Jean-Baptiste LAGRANGE, Christine LE BIHAN, Bernard LE FEUVRE, Marie-Catherine MANENS, Irem de Rennes	
<b><i>Simuler un niveau de chantier et un tachéomètre</i></b>	55
Hubert RAYMONDAUD, Stephan MANGANELLI	
<b><i>Rubrique Vie des Irem</i></b>	62
Consignes pour soumettre un article à Repères Irem	
<b><i>Les « remue-ménages » :</i></b>	
<b><i>motivation des élèves en classe de mathématiques de Seconde par la mise en place d'un rituel de recherche de problèmes ouverts</i></b>	63
Pierre GLASSON, Xavier DURAND, Irem de Grenoble	
<b><i>Rubrique Vie des Irem</i></b>	81
<b><i>PUBLIMATH</i></b>	
<b><i>Rubrique Parutions</i></b>	87
Abonnements, réabonnements	92
Liste des Irem	93
Sommaire du prochain numéro	94

---

## EDITORIAL

---

L'éditorial du numéro 116 promettait que les pages de notre revue resteraient ouvertes aux sciences du numérique. Elles sont un objet de réflexion, de rénovation, de renouvellement ou de préoccupation d'importance : en témoigne le nombre de propositions d'articles qui nous sont parvenues. Le comité d'édition a donc décidé que le numéro 117 leur fasse encore la part belle.

L'article *Trier pour mieux comparer*, du groupe « Informatique sans ordinateur » de l'Irem de Clermont-Ferrand s'appuie sur une expérience proposée en partage. Il sera aussi une source de réflexion sur les relations que mathématiques et informatique peuvent entretenir dans les classes.

Le Groupe Casyopée de l'Irem de Rennes propose dans l'article *Aider à la compréhension des fonctions avec un logiciel* une analyse des apports éventuels de la programmation à la notion de fonction fondée sur une pratique en lycée. Il y apparaît que les « fonctions définies par un algorithme » pourraient constituer un moyen d'en améliorer la compréhension.

Hubert Raymondaut et Stephan Manganelli, enseignants au LEGTA Louis Giraud de Carpentras-Serres, dans leur article *Simuler un niveau de chantier et un théodolite*, nous exposent la manière dont il est possible d'utiliser *GeoGebra* comme support pour un cours de topographie, afin de simuler deux modèles d'appareils.

---

EDITORIAL

---

Enfin la rubrique *multimédia* résume une analyse de Gérard Berry, professeur au Collège de France, qui tente une prospective du développement de l'informatique.

Bien sûr, les mathématiques et leur enseignement restent notre cœur de métier. Un article de Pierre Glasson et Xavier Durand (de l'Irem de Grenoble) consacré aux « Remue-

Méninges », c'est-à-dire à la *Recherche de problèmes ouverts* à travers *la mise en place d'un rituel de recherche*, propose des exemples de situations expérimentées en classe, certaines bien connues, d'autres moins. Une occasion de se (re) motiver ?

Bonne lecture à toutes et à tous !

Alex Esbelin

---

## TRIER POUR MIEUX COMPARER

---

Séverine FLEURY  
Pascal LAFOURCADE  
Marianne MOGNOS  
Malika MORE  
Irem de Clermont-Ferrand

*Résumé* : La notion de comparaison de deux nombres est une notion mathématique enseignée de façon récurrente tout au long du cursus scolaire. En informatique, cette notion est utilisée de façon intensive dans des algorithmes de tri, comme ceux qui permettent, par exemple, d'afficher les articles d'un site de e-commerce par prix croissant. L'objectif de cet article est de proposer une activité mettant en regard ces deux champs. Plus précisément, les élèves se déplacent pour classer des listes de nombres dans l'ordre croissant de manière collaborative en utilisant plusieurs algorithmes. Les nombreuses comparaisons faites pour classer les nombres permettent de s'exercer de manière ludique. Par la même occasion, les élèves découvrent quelques algorithmes de tri classiques, et constatent que certains sont plus efficaces que d'autres.

### 1. — Introduction

La notion de comparaison de deux nombres est travaillée dès le cycle 1 et fait encore partie des programmes de mathématiques des classes de sixième et de cinquième. Les difficultés que rencontrent les élèves pour comparer les nombres relatifs rendent cette notion délicate à enseigner. En effet, il n'est pas naturel pour eux que  $-10$  soit plus petit que  $-4$ . Une autre difficulté apparaît pour comprendre que  $3,14$  est plus grand que  $3,121$ , alors que  $121$  est plus grand que  $14$ . De plus, l'intérêt de savoir comparer deux nombres n'est pas toujours clair pour les élèves.

Pour sa part, l'informatique utilise abondamment les comparaisons dans des algorithmes appelés algorithmes de tri. La nature des

objets comparés et de l'ordre utilisé est variable : des nombres de plusieurs types (entiers, entiers signés, flottants) vis à vis de l'ordre numérique, chaînes de caractères vis à vis de l'ordre alphabétique, objets composés (date, ...) pourvu qu'ils soient munis d'un ordre total (ordre chronologique), etc.

L'idée de l'activité présentée dans cet article est de faire pratiquer la comparaison de nombres aux élèves en leur faisant vivre des algorithmes de tri de manière collaborative. Pour cela, chaque élève joue le rôle d'un nombre parmi une liste. Cette approche concrète et ludique permet également d'initier les élèves aux algorithmes de tri.

Il est à noter que cette activité n'est pas conçue pour être utilisée à des fins de remédiation en ce qui concerne la comparaison de nombres.

Dans la partie 2., les cinq algorithmes de tri utilisés dans l'activité pour la classe sont présentés dans leur principe, puis détaillés dans un exemple, et enfin des consignes permettant de les mettre en œuvre concrètement sont proposées. Dans la partie 3., l'activité pour la classe est détaillée, ainsi que les documents proposés aux élèves. Cette activité a été testée en classes de sixième et cinquième, ainsi qu'en CM2. Enfin, dans la partie 4., nous revenons sur les enjeux informatiques des algorithmes de tri.

## 2. — Algorithmes de tri

Dans la plupart des jeux de cartes (tarot, belote, rami, ...), les joueurs ont plusieurs cartes dans leur main. Afin de mieux les voir et de pouvoir optimiser leur stratégie, ils les ordonnent souvent par couleur et par valeur, en fonction du jeu auquel ils jouent. Comment procèdent-ils pour les ranger ? Il n'y a pas seulement une technique possible. Certains joueurs sont plus rapides que d'autres pour ranger leur jeu. Quel est leur secret ?

Plus sérieusement, classer les données est une activité qui occupe de nombreuses administrations depuis longtemps. Historiquement, le dépouillement des recensements aux USA a été l'une des premières tâches automatisées grâce à une machine. Répondant à un concours du gouvernement américain, Herman HOLLERITH réalisa la première machine à statistiques en 1890. Cette machine ordonne automatiquement des cartes perforées. Par la suite, de grandes compagnies comme IBM proposèrent plusieurs modèles, comme par exemple le modèle D3, qui pouvait traiter 1 000 cartes par minute. La plupart de ces machines utilisent une

méthode de classement appelée *tri radix* aussi appelée *tri par base*, qui n'est pas présentée dans cet article, mais qui permet d'ordonner efficacement des données « structurées », comme des cartes perforées. [1]

Ordonner, ranger, classer un nombre considérable de données sont des opérations fastidieuses pour un être humain, alors qu'un ordinateur est bien plus efficace dans ces tâches répétitives. Il existe de nombreuses manières de programmer un ordinateur pour classer des données. Ainsi, les plus grands informaticiens se sont penchés sur la question afin d'élaborer des algorithmes les plus efficaces possibles [1]. De nos jours, les besoins dans ce domaine sont en constante augmentation, à cause de l'explosion des données disponibles via internet, et un des principaux défis des années à venir est le traitement des données issues du « BIG DATA » et des objets connectés.

Une motivation plus mathématique est le calcul de la médiane d'une suite de valeurs, qui est immédiate sur une liste ordonnée. Cependant, il existe aussi, pour le calcul de la médiane, un algorithme naïf, moins efficace, qui consiste à retirer le maximum et le minimum de la liste, jusqu'à ce qu'il ne reste plus qu'un ou deux éléments.

Le vocabulaire utilisé en informatique est différent de celui utilisé en mathématiques : en informatique, il s'agit de **trier** des nombres dans l'ordre croissant, ou des mots dans l'ordre alphabétique, ou des messages par date, ou des articles par prix croissant, ... alors qu'en mathématiques, ce sont les termes « ordonner » ou « ranger » qui sont habituellement utilisés. Les algorithmes permettant de réaliser ces opérations s'appellent ainsi des *algorithmes de tri*. Ils sont très nombreux : par exemple, la page qui leur est consacrée sur Wikipedia ([https://fr.wikipedia.org/wiki/Algorithme\\_de\\_tri](https://fr.wikipedia.org/wiki/Algorithme_de_tri)) n'en recense

pas moins de dix-neuf différents. L'étude de quelques-uns de ces algorithmes de tri fait partie des incontournables d'un cours d'algorithmique pour les étudiants débutants en informatique, en particulier parce qu'ils sont omniprésents en informatique.

Les algorithmes de tri utilisés dans cette activité sont présentés ci-dessous : *le tri par insertion*, *le tri par minimum* (ou *par maximum*) et *le tri fusion* pour la partie de l'activité réalisée en cycle 3, et *le tri à bulles* et *le tri rapide* pour la partie de l'activité réalisée en cycle 4. Ils sont parmi les plus classiques des algorithmes de tri existants. Chaque algorithme sera décrit comme suit : la présentation intuitive de son fonctionnement ; des consignes permettant de l'expérimenter concrètement ; puis un exemple à partir de la liste :

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

à ranger dans l'ordre croissant

11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912.

Les fiches-élèves de l'activité proposée contiennent des schémas représentant ces algorithmes. Il a été volontairement évité d'écrire les algorithmes sous forme de pseudo-code, pour choisir des représentations imagées, inspirées des fiches IDEA (<https://idea-instructions.com/quick-sort/>), afin de rendre l'activité accessible aux plus jeunes et de permettre une reformulation des algorithmes par les élèves. Pour une présentation classique des algorithmes de tri, le lecteur est invité à consulter [1], par exemple.

### 2.1. Tri par insertion

Cet algorithme simple correspond à ce que font certains joueurs de cartes en prenant les cartes une à une et en les rangeant au fur et à mesure à la bonne place dans leur main. Le nom de « tri par insertion » fait référence à son principe de fonctionnement : partant d'un ensemble

non ordonné de cartes, chaque nouvelle carte est correctement insérée en fonction des cartes déjà présentes.

*Expérience* : Comme cela a déjà été dit, c'est le tri du joueur de cartes. Pour l'expérimenter, il faut un tas de dix cartes portant des nombres, – par exemple dix nombres entiers distincts –, mélangées sur la table, face cachée. Ensuite, il s'agit d'aligner ces cartes, face cachée, en les tirant une par une, de manière à toujours avoir les nombres rangés dans l'ordre croissant. Pour déterminer où insérer une nouvelle carte, il faut retourner chaque carte et la comparer à cette nouvelle carte. C'est un point important, qui a volontairement été négligé dans l'exemple ci-dessous, pour simplifier. Ces consignes devraient induire naturellement l'utilisation de l'algorithme de tri par insertion.

Comme cela a déjà été dit, c'est le tri du joueur de carte, qui consiste à ranger les cartes dans l'ordre croissant sans que jamais plus de deux cartes ne soient retournées (on suppose évidemment que l'on a oublié le nombre écrit sur une carte une fois qu'elle est cachée).

Dans la suite, l'expression « carte  $n$  » désigne la carte en position  $n$  sur l'instant. L'algorithme est alors le suivant :

- Insertion de la carte 2 : retourner les cartes 1 et 2 et les permuter si nécessaire. Cacher les cartes 1 et 2. L'insertion est terminée.
- Insertion de la carte 3 : retourner les cartes 2 et 3.
  - Si les cartes 2 et 3 sont bien rangées, l'insertion est terminée.
  - Sinon, permuter les cartes 2 et 3 puis cacher la carte 3. Retourner la carte 1 et comparer à la carte 2. Les permuter si nécessaire. Cacher les cartes 1 et 2. L'insertion est terminée.



---

 TRIER POUR  
 MIEUX COMPARER
 

---

- Insertion de la carte 4 : retourner les cartes 3 et 4. etc.

*Exemple* : Etant donné la liste des nombres suivants : 666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92, les différentes étapes de l'exécution de cet algorithme sont présentées ci-dessous. Les nombres en gras correspondent à l'élément courant (c'est-à-dire celui qui est en train d'être traité). Les nombres en italique sont les nombres déjà traités. Au début, le premier nombre est seul donc il est forcément bien positionné. Ensuite, un nouveau nombre est inséré à sa place dans la liste des nombres déjà traités, comme c'est le cas, par exemple, dans les lignes 6 et 7, lorsque le nombre **11** est inséré dans la liste triée 42 ; 77 ; 666. Cette étape est répétée jusqu'à ce que tous les nombres aient été insérés à leur place, et la liste finalement obtenue est alors triée. Chacune des sept insertions nécessaires ici est traitée en deux lignes, même si aucune permutation n'est nécessaire.

*État initial* :

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

**666** 42 77 11 912 136 26 92  
 666 **42** 77 11 912 136 26 92  
 42 666 **77** 11 912 136 26 92  
 42 666 **77** 11 912 136 26 92  
 42 **77** 666 11 912 136 26 92  
 42 77 666 **11** 912 136 26 92  
**11** 42 77 666 912 136 26 92  
 11 42 77 666 **912** 136 26 92  
 11 42 77 666 **912** 136 26 92  
 11 42 77 666 912 **136** 26 92  
 11 42 77 **136** 666 912 26 92  
 11 42 77 136 666 912 **26** 92  
 11 **26** 42 77 136 666 912 92  
 11 26 42 77 136 666 912 **92**  
 11 26 42 77 **92** 136 666 912

*État final* :

11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912

## 2.2. Tri par minimum (ou maximum)

L'algorithme de tri par minimum consiste à déterminer le minimum parmi les nombres proposés et à le placer à gauche de la liste en l'échangeant avec le nombre en première position. Cet élément est alors à sa place définitive. Il s'agit ensuite de recommencer avec les nombres restants. En répétant ce processus autant de fois qu'il y a de nombres, nous obtenons les nombres triés du plus petit au plus grand. Il est possible de faire la même chose en cherchant le maximum à chaque fois, et en le plaçant à droite. Cet algorithme est parfois aussi appelé tri par sélection, puisqu'il consiste à sélectionner le minimum (ou le maximum) des éléments restant à traiter.

*Expérience* : Il faut placer huit cartes portant des nombres, par exemple des nombres entiers distincts, mélangées sur la table. Ces cartes forment un tas face cachée. Il faut aussi une rangée de huit emplacements initialement vides destinés à recevoir la liste triée, un emplacement où stocker une carte au choix, face visible, et une pile destinée à recevoir les cartes non encore triées. Il n'est possible de prendre qu'une carte à la fois sur le tas et la placer : soit dans l'emplacement, soit dans la rangée à sa place définitive, soit sur la table dans une seconde pile face cachée.

Comment trouver le minimum des huit cartes en ne voyant qu'une seule fois chaque carte ? La recherche du minimum est une étape importante de l'algorithme, mais qui a été volontairement négligée dans l'exemple ci-dessous pour simplifier.

*Solution* : Il faut mettre la première carte de côté et parcourir les autres une à une en les comparant à la première, puis les intervertir dès que la carte est plus petite que celle mise de côté.

*Exemple* : Étant donné la liste des nombres suivants : 666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92, les différentes étapes de l'exécution de cet algorithme sont présentées ci-dessous. Les nombres en gras correspondent à l'élément courant qui est en train d'être traité, soit le minimum des éléments restant à traiter. Les nombres en italiques sont les nombres déjà traités. À chaque étape, le nombre en cours de traitement est placé à sa place définitive, c'est-à-dire à droite de la liste des éléments déjà traités, en l'échangeant avec l'élément qui s'y trouvait. C'est ce qui se passe ci-dessous, dans les lignes 5 et 6, pour le nombre **42** qui est placé à droite de la liste triée *11, 26*, en l'échangeant avec le nombre 77.

*État initial* :

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

666 42 77 **11** 912 136 26 92

—> 11 est le minimum des éléments restants à traiter

*11* 42 77 666 912 136 26 92

—> 11 est échangé avec 666 pour être à sa place définitive

*11* 42 77 666 912 136 **26** 92

*11 26* 77 666 912 136 42 92

*11 26 77* 666 912 136 **42** 92

*11 26 42* 666 912 136 **77** 92

*11 26 42 77* 912 136 666 92

*11 26 42 77 92* 136 666 **92**

*11 26 42 77 92* 136 666 912

*11 26 42 77 92* 136 **666** 912

*11 26 42 77 92* 136 **666** 912

*11 26 42 77 92* 136 **666** 912

*11 26 42 77 92* 136 **666** 912

*11 26 42 77 92* 136 666 **912**

*11 26 42 77 92* 136 666 **912**

*État final* :

*11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912*

### 2.3. Tri fusion

Sur des listes suffisamment longues, l'algorithme de tri fusion est plus rapide que les deux précédents. Il repose sur les deux remarques suivantes :

- une liste composée d'un seul élément est nécessairement ordonnée.
- il est facile de fusionner en une liste ordonnée deux listes déjà ordonnées. Pour cela il suffit de comparer la première carte de chaque liste, de ranger la plus petite des deux dans une nouvelle liste, et de recommencer sur ce qui reste des deux listes initiales. Ce processus s'appelle *l'interclassement*.

L'algorithme de tri fusion consiste à décomposer la liste initiale en éléments isolés, puis à interclasser ces listes d'un élément en listes triées de deux éléments, puis à interclasser les listes triées de deux éléments en listes triées de quatre éléments, et ainsi de suite jusqu'à obtenir une seule liste triée.

*Expérience* : C'est une version à plusieurs participants. Il faut  $n$  personnes pour trier  $n$  cartes. Comme précédemment, chaque carte comporte un nombre entier distinct des autres. Chaque participant possède une carte (qui forme une liste triée à elle toute seule). Chacun choisit un autre participant et ils construisent ensemble une liste de deux cartes triée par interclassement. Ensuite, chaque groupe fusionne avec un autre (c'est-à-dire interclasse ses cartes avec celles d'un autre groupe), et ainsi de suite jusqu'à ce qu'il n'y ait plus qu'un seul groupe.

*Exemple* : Étant donné la liste des nombres suivants : 666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92, les différentes étapes de l'exécution de cet algorithme sont présentées ci-dessous.

TRIER POUR  
MIEUX COMPARER

*État initial :*

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

huit nombres isolés :

| 666 | 42 | 77 | 11 | 912 | 136 | 26 | 92 |

quatre listes triées de deux nombres :

| 42 666 | 11 77 | 136 912 | 26 92 |

deux listes triées de quatre nombres :

| 11 42 77 666 | 26 92 136 912 |

une liste triée de huit nombres :

| 11 26 42 77 92 136 666 912 |

*État final :*

11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912

#### 2.4. Tri à bulles

Le principe du tri à bulles est de parcourir la liste en échangeant, lors du parcours, deux éléments consécutifs s'ils sont rangés dans l'ordre inverse, et de répéter ce processus jusqu'à ce qu'il n'y ait plus d'échanges lors d'un parcours. Dans ce cas, la liste est triée.

Une variante du tri à bulles est le *tri shaker* qui consiste, au lieu de revenir au début de la liste, à effectuer un parcours dans le sens inverse, en effectuant les échanges selon l'ordre inverse. Le nom de cet algorithme de tri vient du fait que les plus grands éléments « montent » vers la droite de la liste pendant les parcours, comme des bulles dans un liquide.

*Expérience :* Il faut aligner huit cartes portant des nombres, par exemple des nombres entiers distincts, mélangées sur la table, face cachée. Il n'est possible de retourner que deux cartes consécutives et uniquement les inverser ou pas, avant de les retourner à nouveau. Il n'est pas non plus possible de mémoriser la valeur des cartes déjà retournées. Comment trier les cartes ?

*Exemple :* Étant donné la liste des nombres suivants : 666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92, les différentes étapes de l'exécution de cet algorithme sont présentées ci-dessous. Les couples de nombres échangés sont indiqués en gras. À la fin du premier parcours, le plus grand élément de la liste est forcément à sa place définitive, à la fin du second parcours, c'est aussi le cas de l'avant dernier élément, et ainsi de suite. Les éléments rangés à leur place définitive sont indiqués en italique.

*État initial :*

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

**666 42** 77 11 912 136 26 92

**42 666** 77 11 912 136 26 92

42 **77 666** 11 912 136 26 92

42 77 **666 11** 912 136 26 92

42 77 **11 666** 912 136 26 92

42 77 11 **666 912** 136 26 92

42 77 11 666 **912 136** 26 92

42 77 11 666 **136 912** 26 92

42 77 11 666 136 **912 26** 92

42 77 11 666 136 **26 912** 92

42 77 11 666 136 26 **912 92**

42 77 11 666 136 26 **92 912**

42 77 11 666 136 26 92 **912**

—> fin du premier parcours : l'élément 912 est à sa place définitive

**42 77** 11 666 136 26 92 *912*

—> début du deuxième parcours

42 **77 11** 666 136 26 92 *912*

42 **11 77** 666 136 26 92 *912*

42 11 **77 666** 136 26 92 *912*

42 11 77 **666 136** 26 92 *912*

42 11 77 **136 666** 26 92 *912*

42 11 77 136 **666 26** 92 *912*

42 11 77 136 **26 666** 92 *912*

42 11 77 136 26 **666 92** *912*

42 11 77 136 26 **92 666** *912*

42 11 77 136 26 92 **666** *912*

—> fin du deuxième parcours : l'élément 666 est à sa place définitive

42 11 77 136 26 92 666 912

—> début du troisième parcours

11 42 77 136 26 92 666 912

11 42 77 136 26 92 666 912

11 42 77 136 26 92 666 912

11 42 77 136 26 92 666 912

11 42 77 26 136 92 666 912

11 42 77 26 136 92 666 912

11 42 77 26 92 136 666 912

11 42 77 26 92 136 666 912

—> fin du troisième parcours : l'élé-  
 ment 136 est à sa place définitive

11 42 77 26 92 136 666 912

—> début du quatrième parcours

11 42 77 26 92 136 666 912

11 42 77 26 92 136 666 912

11 42 26 77 92 136 666 912

11 42 26 77 92 136 666 912

11 42 26 77 92 136 666 912

—> fin du quatrième parcours : l'élé-  
 ment 92 est à sa place définitive

11 42 26 77 92 136 666 912

—> début du cinquième parcours

11 42 26 77 92 136 666 912

11 26 42 77 92 136 666 912

11 26 42 77 92 136 666 912

11 26 42 77 92 136 666 912

—> ici, le parcours n'amène plus  
 aucun échange, donc l'exécution de  
 l'algorithme est terminée

*État final :*

11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912

### 2.5. Tri rapide (ou tri pivot ou Quicksort)

En 1962, Tony HOARE [2] invente le *tri rapide* (Quicksort). Certains disent qu'il s'agit aujourd'hui de l'algorithme le plus utilisé dans le monde !

L'idée de cet algorithme est de choisir un élément appelé *pivot* et de le ranger à sa place définitive, en plaçant tous les éléments inférieurs au pivot à sa gauche et tous ceux qui sont supérieurs au pivot à sa droite. Plus précisément, cet algorithme fonctionne comme suit :

- choisir un élément  $p$  au hasard parmi les éléments à trier. Cet élément est appelé le pivot.
- séparer les éléments restants en deux listes : d'une part ceux qui sont plus petits que  $p$ , et d'autre part, ceux qui sont plus grands.
- trier chacune des listes obtenues en utilisant ce même algorithme.
- agrèger les deux listes triées, en les mettant bout à bout, et le pivot entre les deux.

*Expérience :* Pour illustrer ce tri, il est possible d'utiliser des boîtes d'allumettes contenant un nombre variable de billes identiques, et une balance à fléau pour comparer leurs masses. La consigne est de prendre au hasard les boîtes une par une, et de les mettre chacune à leur place définitive en les comparant avec les autres, grâce à la balance.

Plus précisément, au départ, toutes les boîtes sont alignées mais non ordonnées :

- choisir une boîte au hasard. Cette boîte sera appelée la boîte pivot.
- mettre à gauche du pivot toutes les boîtes plus légères que la boîte pivot, et à droite toutes les boîtes plus lourdes que la boîte pivot.
- la position du pivot par rapport aux autres boîtes est donc trouvée.

Recommencer l'opération d'une part avec les boîtes de droite et d'autre part avec les boîtes de gauche, ceci jusqu'à avoir trouvé la place de toutes les boîtes.

---

**TRIER POUR  
MIEUX COMPARER**


---

*Exemple* : Étant donné la liste des nombres suivants : 666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92, les étapes de l'exécution de l'algorithme de tri rapide sont décrites ci-dessous, où les pivots, tirés aléatoirement, sont successivement 77, 11, 666, 42, 92 (en gras). Les éléments à leur place définitive sont indiqués en italique.

*État initial* :

666 ; 42 ; 77 ; 11 ; 912 ; 136 ; 26 ; 92

666 42 **77** 11 912 136 26 92

—> le pivot 77 est choisi au hasard

11 42 26 **77** 912 136 92 666

—> tous les nombres rangés à gauche de 77 sont plus petits que lui et tous ceux placés à droite sont plus grands

11 42 26 77 912 136 92 666

—> le pivot 11 est choisi au hasard dans la 1<sup>e</sup> sous-liste

11 42 26 77 912 136 92 666

11 42 26 77 912 136 92 **666**

—> le pivot 666 est choisi au hasard dans la 2<sup>e</sup> sous-liste

11 42 26 77 136 92 **666** 912

11 **42** 26 77 136 92 666 912

—> le pivot 42 est choisi au hasard

11 26 **42** 77 92 136 666 912

11 26 42 77 **92** 136 666 912

—> le pivot 92 est choisi au hasard

11 26 42 77 **92** 136 666 912

*État final* :

11 ; 26 ; 42 ; 77 ; 92 ; 136 ; 666 ; 912

Il existe de nombreuses ressources pédagogiques sur les algorithmes de tri, comme par exemple ce site, où il est possible de voir le déroulement de nombreux algorithmes de tri sur des exemples : [https://interstices.info/jcms/c\\_6973/les-](https://interstices.info/jcms/c_6973/les-)

[algorithmes-de-tri?hlText=tri](http://www.sorting-algorithms.com/). Le site suivant (en anglais) propose de nombreuses références aux articles originaux sur la complexité des algorithmes de tri et une visualisation permettant de comparer l'efficacité de différents algorithmes : <http://www.sorting-algorithms.com/>.

Enfin, le site suivant propose des danses folkloriques basées sur des algorithmes de tri : <http://algo-rythmics.ms.sapientia.ro/>.

### 3. — Déroulé de l'activité

#### 3.1. Cadre de l'expérimentation

Cette activité a été expérimentée dans un collège situé en zone rurale par l'une des co-auteurs avec sa classe de sixième en 2017-2018. Ces élèves ont l'habitude de tester des activités du groupe IREM-MPSA « Faire de l'informatique sans ordinateur ». La seconde partie de l'activité a aussi été menée avec ces mêmes élèves, passés en cinquième, en début d'année scolaire 2018-2019.

De plus, au cours de cette rentrée 2018-2019, l'activité a de nouveau été testée dans deux classes de sixième d'un autre collège de même profil rural, établissement dans lequel enseigne l'une des co-auteurs.

Enfin, la première partie de cette activité (tris par minimum, par insertion et fusion) a aussi été testée en octobre 2018 dans une classe de CM2, avec des groupes de 9 élèves (8 et le chef d'orchestre). Chaque groupe a travaillé sur un seul algorithme de tri.

Dans les parties 3.5. et 3.6., un compte-rendu des expériences sur le terrain est exposé, ainsi qu'un certain nombre d'observations et de propositions d'améliorations de l'activité.

### 3.2. Objectifs

Le premier objectif est de faire le lien entre les mathématiques, où les élèves sont amenés à comparer des nombres de toutes sortes, et l’informatique, où le tri de données a beaucoup d’importance. Les élèves comprennent que leur usage des ordinateurs et d’internet implique qu’ils sont nécessairement utilisateurs réguliers des algorithmes de tri, même si ce n’est pas explicite.

Ensuite, les objectifs informatiques sont de découvrir qu’il existe plusieurs façons de trier, que l’on peut comparer des algorithmes et que certains peuvent être plus efficaces que d’autres selon les situations, les données à traiter, etc.

Enfin, l’objectif mathématique est d’utiliser le vocabulaire vu en cours (ranger dans l’ordre croissant/décroissant, comparer, intercaler, encadrer, inférieur à, supérieur à) en le

faisant vivre et en reformulant les schémas des fiches-élèves, afin que les élèves l’assimilent mieux.

De plus, il est toujours intéressant de montrer qu’une même consigne peut être traitée par différentes méthodes. Le professeur pourra insister sur le fait que les élèves mixent souvent ces différents algorithmes pour ordonner une liste de nombres. En outre, à l’heure où la programmation fait partie intégrante des programmes de mathématiques, c’est aussi l’occasion d’étudier des algorithmes incontournables.

### 3.3. Matériel

Les fiches-élèves sont présentées ci-après.

Les consignes suivantes apparaissent sur chaque fiche-élève, en plus des explications pour chaque algorithme de tri.

#### CONSIGNES :

- En observant chaque étape, comprendre le fonctionnement de ce tri :  
Quels sont les disques pointés par le chef d’orchestre ? Quelle question se pose-t-il à leur propos ? Selon la réponse, quelles actions leur demande-t-il d’effectuer (changement de place ou non, pour aller où) ?
- A la fin, les disques doivent être rangés du plus petit au plus grand.
- Choisir un chef d’orchestre qui va trier les autres membres du groupe (par exemple, faire choisir un nombre entier à chacun entre 1 et 100).

#### OBJECTIF FINAL :

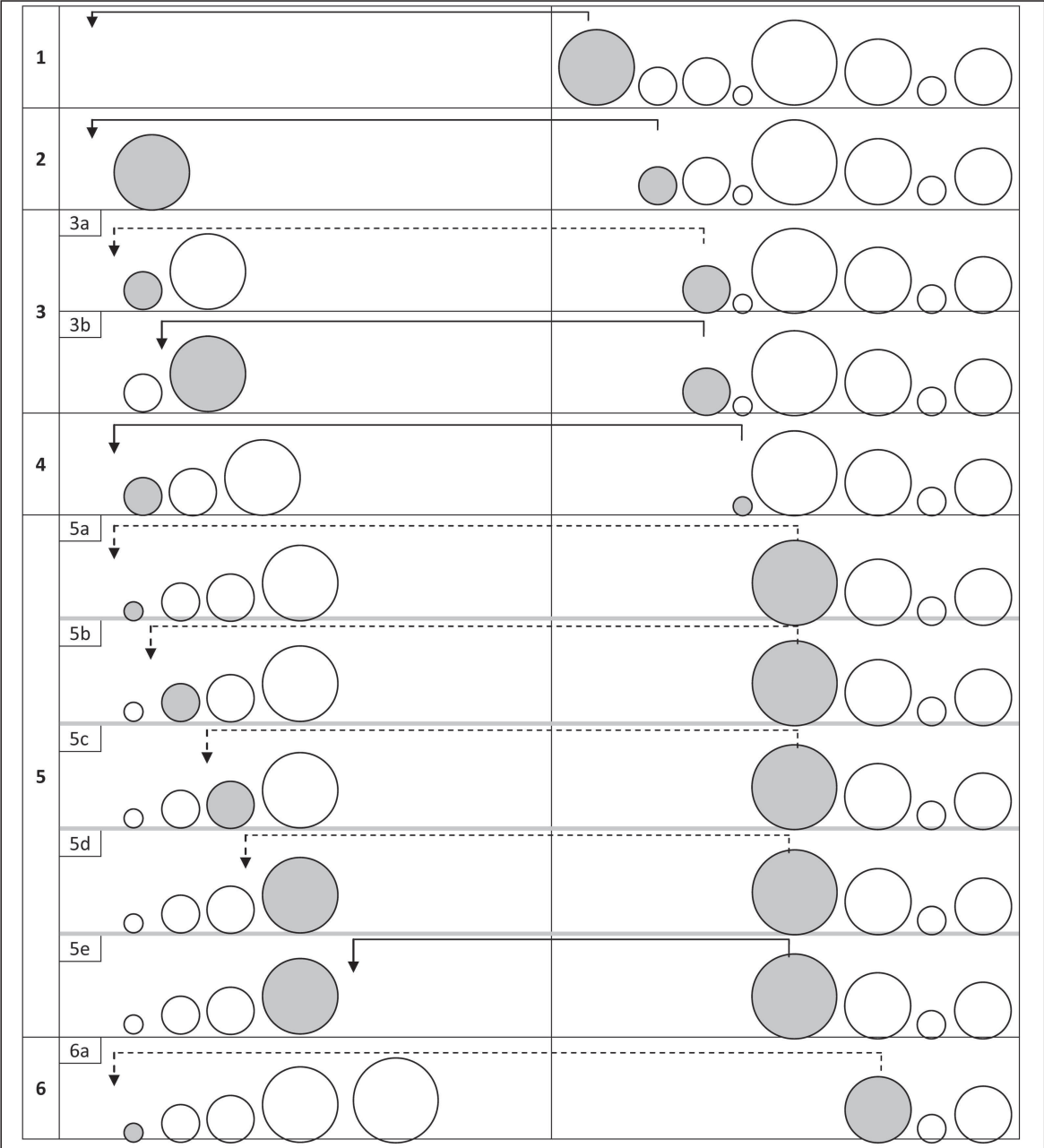


#### LEGENDE :

- pas trié      ● trié      ◐ pointé par le chef d’orchestre

TRIER POUR  
MIEUX COMPARER

**TRI PAR INSERTION (TRI DU JEU DE CARTES)**



*Principe du tri par insertion :*  
Prendre les disques un par un et les ranger correctement à leur place au fur et à mesure.

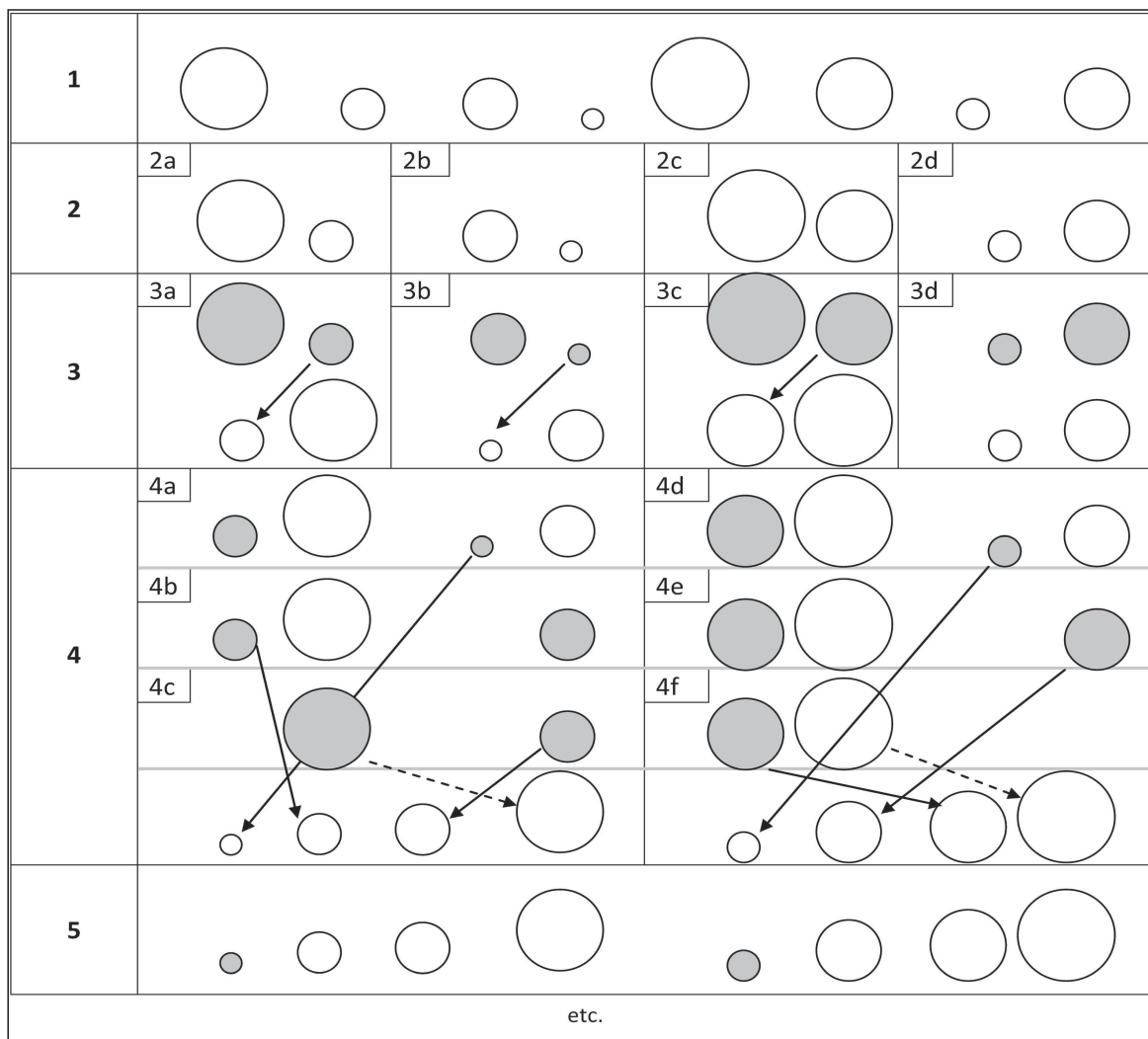
**TRIER PAR MINIMUM (TRI SELECTION)**

1	1a	
	1b	
	1c	
	1d	
	1e	
	1f	
	1g	
	1h	
2	2a	
	2b	
	2c	
	2d	
		etc.

*Principe du tri par minimum :*  
 Identifier le plus petit parmi les disques et le placer au début de la liste.  
 Recommencer avec les disques restants.



**TRI FUSION**



*Principe du tri fusion :*

La première étape est d'interclasser deux listes d'un seul élément. On obtient alors des listes triées de deux éléments.

Pour la deuxième étape, procéder de la manière suivante : comparer les deux premiers disques, puis ranger le plus petit dans une nouvelle liste, et recommencer avec les disques restants. On obtient alors des listes triées de quatre éléments.

Recommencer comme la deuxième étape avec ces listes triées de quatre éléments, puis celles de huit, etc.

### TRI A BULLES

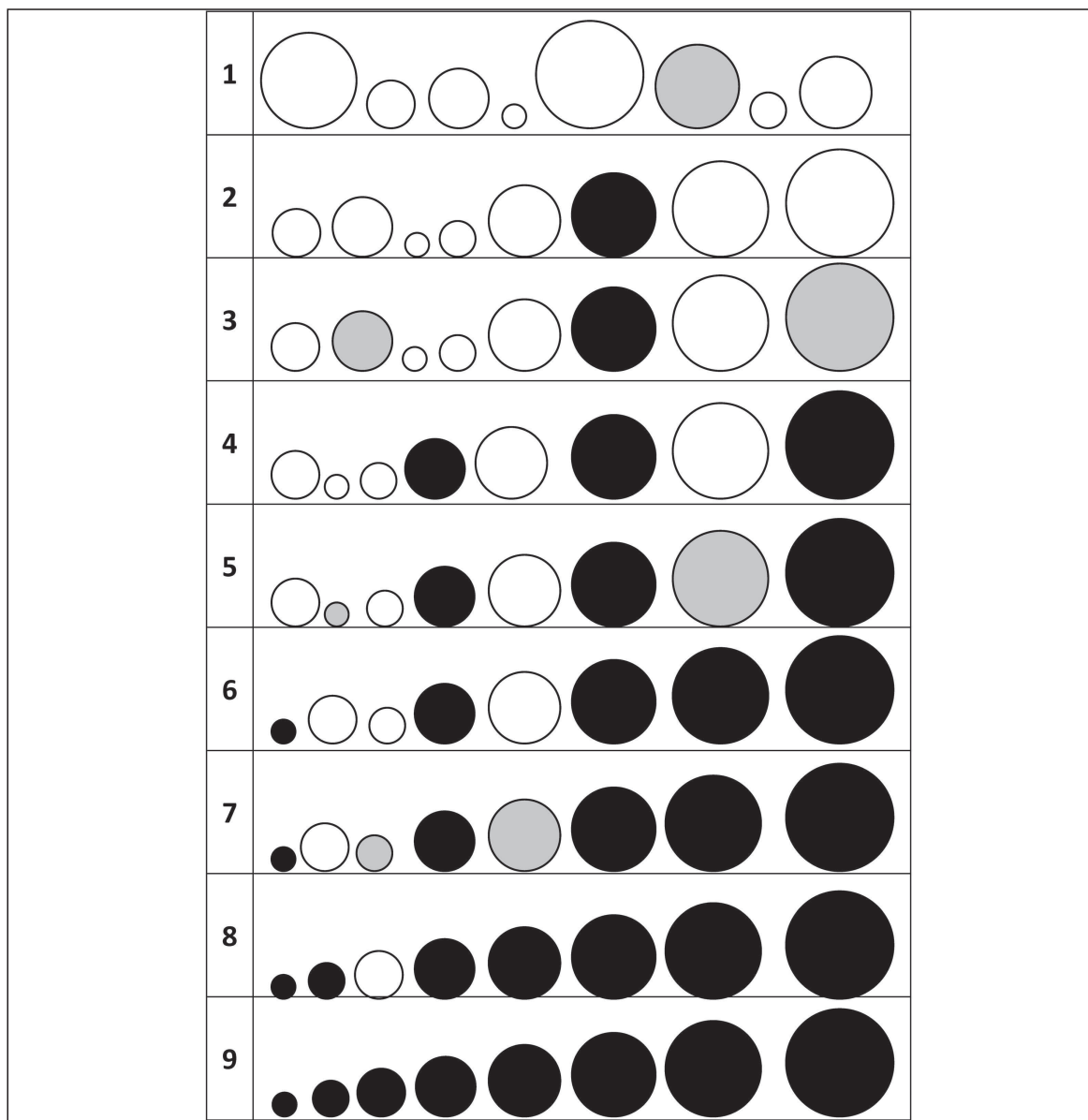
1	1a	
	1b	
	1c	
	1d	
	1e	
	1f	
	1g	
	1h	
	1i	
	1j	
	1k	
	1l	

etc.

*Principe du tri à bulles :*

Parcourir la liste de disques en échangeant deux disques consécutifs rangés dans le mauvais ordre. Recommencer jusqu'à ce qu'il n'y ait plus d'échanges possibles.

**TRI RAPIDE (TRI PIVOT)**



*Principe du tri rapide :*

Choisir un disque « pivot » et le ranger à sa place définitive, en triant tous les disques plus petits à sa gauche et tous les disques plus grands à sa droite.

Recommencer en choisissant un nouveau disque « pivot » dans chaque paquet nouvellement créé, et jusqu'à ce que tous soient triés.

### 3.4. Déroulement prévu de l'activité

C'est une activité « deux en une » en quelque sorte, puisque le déroulement est le même selon que le choix soit de travailler les tris par minimum, par insertion et fusion (plutôt en sixième, voire en cycle 3) ou les tris à bulles et rapide (plutôt en cinquième, voire en cycle 4). Tout d'abord, le professeur échange avec les élèves sur le fait qu'en informatique, l'ordinateur est en permanence amené à classer (tri par prix croissant sur des sites de e-commerce par exemple, tri par date/taille/ordre alphabétique sur les dossiers qu'il a sur sa clé, ...), même quand l'utilisateur ne s'en aperçoit pas. Il revient aussi sur le fait qu'une machine ne peut pas avoir une vue d'ensemble des données, et qu'elle doit les traiter par parquets de petite taille fixée. Il explicite que pour le tri, les opérations de base étant les comparaisons et les permutations, la machine doit traiter deux données à la fois et qu'elle pourra s'en contenter. Cela expliquera pourquoi, par la suite, presque toutes les données sont cachées.

Dans un deuxième temps, le professeur forme des groupes (deux ou trois selon le nombre d'algorithmes de tri à étudier) et demande aux élèves de comprendre le fonctionnement de l'un d'eux à l'aide des fiches. Il prévoit plusieurs exemplaires de chaque fiche afin que les élèves puissent travailler en petit nombre s'ils en ressentent le besoin.

Suite à cette phase d'appropriation, les élèves d'un même groupe choisissent un chef d'orchestre parmi eux, puis chacun un nombre. Les élèves-nombres (chacun est le seul à connaître le nombre qu'il a choisi) se placent en ligne devant le chef d'orchestre et ne doivent bouger qu'en suivant ses instructions. Le chef d'orchestre demande alors à ses camarades de se déplacer en avant ou en arrière, de changer de place, de se retourner, de s'accroupir, ... de

manière à exécuter l'algorithme de tri étudié. Si le temps le permet, cette phase d'entraînement peut être répétée en changeant le chef d'orchestre, le but étant que chaque élève du groupe maîtrise au mieux l'algorithme.

Dans la dernière étape de cette activité, qui est facultative, le professeur désigne un chef d'orchestre dans chacun des groupes et impose la même liste de nombres aux élèves de tous les groupes, qui garderont la même position initiale. Les différents algorithmes de tri sont alors comparés en chronométrant le temps mis pour obtenir la liste triée.

### 3.5. Choix pédagogiques

*Place dans les progressions en mathématiques* : Selon l'objectif premier de l'enseignant, cette activité peut être menée avant le chapitre sur la comparaison ou après. En effet, la notion de « tri » en informatique est liée à plusieurs notions en mathématiques : comparer, intercaler, encadrer, ranger dans l'ordre croissant/décroissant. Dès lors, cette activité peut être une introduction pour découvrir l'intérêt de trier, ou bien une conclusion en guise de réinvestissement.

*Choix des nombres* : En sixième (ou en cycle 3), les algorithmes de tri s'effectueront sur des nombres entiers, le principal objectif étant d'en comprendre le fonctionnement et de s'approprier l'activité. L'enseignant fera vivre le vocabulaire mathématique de la comparaison et fera le lien avec le tri informatique.

En cinquième (ou en cycle 4), les nouveaux algorithmes de tri pourront être faits d'abord sur des nombres entiers, puis des nombres décimaux, pour finir sur des nombres relatifs en fin d'année. L'activité peut être reprise à différents moments de l'année et apparaître comme un fil conducteur.

*Choix des algorithmes de tri* : Sur les cinq proposés, trois le sont en sixième et deux en cinquième. La première raison de ce choix est la compréhension et l'appropriation par les élèves. Il s'avère que leur intuition se porte plus souvent sur le tri par insertion, ou par minimum ou le tri à bulles. Ainsi, il est intéressant de faire apparaître petit à petit les deux autres afin de bien les comprendre et les appliquer.

La deuxième raison est l'efficacité des algorithmes de tri. En effet, le tri fusion et le tri rapide d'une part, et les trois autres d'autre part, étant relativement équivalents, il est plus intéressant de comparer des algorithmes de tri d'efficacités différentes, afin de pouvoir ouvrir le débat avec les élèves.

*Choix de la mise en page des fiches distribuées aux élèves* : La conception des fiches a été pensée dans le but que les élèves les plus jeunes se les approprient et puissent reformuler les algorithmes à leur manière, notamment au cycle 3 et pas forcément en sixième. Les schémas ont remplacé un long texte, où le vocabulaire aurait pu poser problème. Les élèves essaient de comprendre le fonctionnement de l'algorithme sans l'aide de l'enseignant, dans un premier temps. Ils échangent avec leurs propres mots ce qu'ils comprennent des schémas, puis font des essais, le professeur restant à l'écoute de leur bonne compréhension.

*Choix du nombre d'élèves* : Le nombre d'élèves est un facteur à prendre en compte. En effet, s'il est trop petit (inférieur à dix), il n'y aura que peu d'étapes et les algorithmes paraîtront très rapides. Il sera alors difficile de pouvoir les comparer et cela limitera leur compréhension, puisque les différents cas possibles pourront ne pas apparaître.

### 3.6. Bilan et constats

C'est une activité qui a du succès auprès des élèves pour plusieurs raisons. La première est qu'elle est proposée sous forme de travail en groupe et basée sur la compréhension des fiches en autonomie. C'est un facteur motivant pour les élèves, qui se sentent plus responsables et au centre de l'apprentissage. La seconde est que les élèves s'approprient mieux les différents algorithmes de tri, du fait qu'ils les « vivent » réellement. Dans le premier collège, où les élèves étaient familiers des activités du groupe « Faire de l'informatique sans ordinateur », cette activité a été menée en début d'année de sixième, sur une séance d'une heure. A la fin de l'année, ce sont les élèves eux-mêmes, sans les fiches sous les yeux, qui ont souhaité « rejouer » ces algorithmes de tri, de mémoire. Et en début de cinquième, lorsqu'ils ont eu à travailler sur les tris à bulles et fusion, l'activité s'est encore mieux déroulée, du fait qu'ils connaissaient déjà les règles.

Dans le deuxième collège, les élèves n'avaient pas encore l'habitude de ce genre d'activité, car elle a été testée en début d'année de sixième. La professeure a fait le choix de faire l'activité en demi-classe et d'expérimenter l'algorithme après l'algorithme pour mieux gérer les questions et mieux observer les élèves en action. Elle a commencé par le tri minimum avec quatorze élèves.

Dans un premier temps, un échange professeur-élèves a eu lieu sur le fait que les ordinateurs classent des données en permanence. Les élèves ont eu différentes idées et l'échange a été riche. Ils n'ont pas forcément pensé au site de e-commerce car cela ne faisait pas partie de leur pratique courante. Puis les élèves ont réfléchi à la compréhension du tri par minimum en petits groupes de quatre ou cinq. Au moment

de la mise en commun, ils ont fini par dire qu'ils ne comprenaient pas. Le professeur a alors utilisé la fiche projetée au tableau pour leur demander ce qu'il se passait à chaque étape et au fur et à mesure. Ils ont formulé le rôle du chef d'orchestre. Le vocabulaire employé a alors rejoint celui des mathématiques, le verbe revenant le plus souvent étant « comparer ». Pour le tri par insertion, ils ont de nouveau réfléchi en petits groupes pour ensuite mettre en commun. Se basant sur ce qui avait été fait sur l'algorithme précédent, ils ont réussi à le faire fonctionner en montrant les différentes étapes de la fiche et en formulant le rôle du chef d'orchestre. L'exécution de l'algorithme s'est ensuite très bien passée. Elle s'est révélée plus rapide que dans le cas précédent (les élèves ont gardé les mêmes nombres). Par manque de temps lors de cette séance de 30 minutes, le tri fusion n'a pas été abordé.

Par la suite, l'activité a été refaite dans des groupes de 10-12 élèves avec le même procédé et en testant en plus, cette fois-ci, le tri fusion. Les tris par minimum et par insertion se sont à nouveau bien déroulés. Des élèves ont ressenti le besoin d'écrire des nombres dans les disques pour rendre la fiche moins abstraite. Le tri fusion a nécessité des explications orales de la part du professeur pour formuler les deux premières étapes. Le test s'est bien déroulé même si le professeur est davantage intervenu pour guider le chef d'orchestre. Les algorithmes n'ont pas été testés en parallèle pour être comparés. Le bon comportement des élèves, leur intérêt et leur investissement ont été encourageants. Ils ont été ravis d'apprendre qu'ils feraient d'autres activités de ce type dans l'année.

Toutefois, nous avons aussi constaté certaines limites à cette activité. En effet, elle est assez efficace d'un point de vue informatique, mais d'un point de vue mathématique, sans surprise, l'activité ne permet pas à elle seule de

surmonter l'erreur des élèves qui comparent mal 3,14 et 3,121. Il faut aussi être vigilant sur les différentes représentations qui peuvent poser problème : les fiches comparent des disques, tandis qu'il est demandé aux élèves de comparer des nombres. Et enfin, il faut aussi tenir compte du fait qu'il ne doit pas y avoir seulement le chef d'orchestre qui comprenne l'algorithme. Tous les élèves du groupe doivent bien le comprendre, en même temps qu'ils doivent retenir le nombre qu'ils ont choisi, sans en changer. Rester bien concentré est alors primordial mais ne suffit pas toujours. Il est proposé, dans la partie suivante, quelques solutions pour pallier ces difficultés.

D'autre part, l'étape de compétition dans cette activité n'a pas été probante, du fait notamment du nombre d'élèves insuffisant dans chaque groupe, du problème de cadencement des actions, et parce que ce qui est à mesurer n'est pas explicite.

Enfin, avec les élèves de CM2, la phase de compétition a été suivie d'une mise en commun. Chaque groupe a présenté son algorithme de tri en action aux autres élèves, qui ont eu à reformuler les étapes pour attester de leur bonne compréhension.

### 3.7. Retour des expériences du terrain

Cette activité s'est révélée aussi enrichissante pour nous, enseignantes, que pour nos élèves. Certains ont montré des qualités insoupçonnées, grâce à l'autonomie proposée dans ce travail de groupe. Nous avons aussi constaté que c'est une manière ludique et efficace d'explicitier des notions et des algorithmes qu'ils utilisent, pour certains, déjà, comme le tri par insertion et le tri minimum. La découverte d'autres algorithmes de tri éveille leur curiosité et aiguise leur motivation. Certes, il faut prévoir de la place, afin que chaque groupe puis-

---

 TRIER POUR  
 MIEUX COMPARER
 

---

se échanger facilement, et aussi moduler le volume sonore afin que celui-ci n'augmente pas trop sous l'effet de l'entrain et de l'élan des élèves. Le temps d'appropriation des fiches en petits groupes au préalable et le fait de projeter la fiche à ce moment-là sont des paramètres dont il faut vraiment tenir compte. Malgré tout, c'est une activité efficace du point de vue des objectifs informatiques et mathématiques, qui convoque aussi de nombreuses qualités et compétences humaines : respect, écoute, échange, ...

Dans le deuxième collège, deux semaines après l'activité, il a été proposé aux élèves de rédiger une affiche sur chaque algorithme de tri, qui en explique rapidement le principe, afin de pouvoir vérifier l'acquisition et la compréhension de ces algorithmes. Cette étape de reformulation permet de conceptualiser les notions informatiques abordées et de réactiver le vocabulaire mathématique (comparer, intercaler, ...). Par la suite, tout au long de l'année, ces affiches pourront être réutilisées dès que l'occasion se présentera et quel que soit le chapitre en cours.

Ainsi donc, pour récapituler, nous proposons quelques astuces pour essayer de solutionner les difficultés rencontrées au fur et à mesure qu'elles apparaissent :

- certains sites de e-commerce sont bloqués par un système de contrôle informatique type proxy ou firewall dans les établissements scolaires. Il est donc nécessaire d'en trouver qui ne soient pas bloqués, pour pouvoir donner un exemple.
- un premier temps en petits groupes permet à chacun d'observer la fiche et d'échanger sur ce qu'ils ont ou non compris.
- la fiche peut être projetée et expliquée à l'ensemble du groupe. Cela permet ainsi au professeur de s'adresser au groupe réuni pour voir si les différentes étapes sont ou non com-

prises. En les pointant, cela permet de montrer les déplacements des différents disques d'une étape à l'autre.

- le professeur peut être le chef d'orchestre pour un premier essai si le groupe n'a pas compris le principe.
- notamment pour le tri fusion, il est préférable de travailler avec un nombre d'élève qui soit une puissance de 2, afin d'éviter que certains restent inactifs trop longtemps.
- l'algorithme de tri peut être exécuté avec des nombres écrits sur papier et disposés sur une table, pour gagner de la place lors de la phase d'entraînement.
- les élèves peuvent écrire des nombres dans les disques sur les fiches afin de surmonter la différence de représentations : disques sur les fiches et nombres en réalité.
- les emplacements peuvent être dessinés au sol afin de mieux distinguer les nombres « pointés » des nombres « déjà traités ».
- il faut prévoir de la place, quitte à déplacer les tables de la salle de classe ou aller dehors, pour bénéficier d'un espace suffisamment grand.

### 3.8. Perspectives

Plus tard dans l'année ou quelques jours après l'activité, un réinvestissement possible serait de proposer aux élèves, de réactiver ces algorithmes de tri avec des étiquettes-nombres à manipuler sur table en binôme. C'est l'occasion de s'assurer notamment que tous les élèves ont bien compris tous les algorithmes. Un objectif mathématique pourrait être alors d'utiliser des nombres décimaux (et/ou relatifs) avec un algorithme maîtrisé par les élèves.

Et pourquoi pas un film qui montrerait un ou plusieurs algorithmes de tri en action ?

Pour des questions de droits à l'image, les élèves pourraient cacher leur visage avec une feuille sur laquelle serait écrit le nombre choisi, ou encore un loup, ... Les films pourraient être ré-exploités l'année suivante afin de servir de support à la place ou en complément des fiches.

Et ne pourrait-on pas trier autre chose que des nombres ? Des mots dans l'ordre alphabétique, des dates, ...

#### 4. — Enjeux informatiques

Comme cela a été dit plus haut, les algorithmes de tri sont omniprésents en informatique, et la connaissance de plusieurs d'entre eux fait partie de la culture générale fondamentale de cette discipline. Avec l'activité proposée dans cet article, les élèves découvrent qu'il existe plusieurs algorithmes pour réaliser cette tâche. C'est très souvent le cas en informatique, et une part importante du travail d'un informaticien consiste à choisir, parmi plusieurs algorithmes possibles, pour un problème donné, celui qui est le mieux adapté au contexte. Lorsque plusieurs algorithmes permettent de réaliser la même tâche, différents critères permettent de les différencier : la facilité plus ou moins grande de mise en œuvre en fonction de la structure des données, l'utilisation plus ou moins efficace de la mémoire pour les calculs, la rapidité d'exécution, etc. Ce dernier critère joue un rôle particulièrement important en algorithmique, sous le nom de « complexité » des algorithmes.

Dans l'activité proposée pour la classe, lorsque les élèves jouent les algorithmes de tri en classe entière, le professeur pourrait les amener à compter les opérations effectuées. Cela pourrait être réalisé par exemple en cadencant les actions par des tops donnés par l'enseignant. Pour comparer entre eux ces algorithmes,

les opérations pertinentes (celles qu'il faut compter) sont les comparaisons de nombres et les échanges de positions. Les élèves constatent ainsi que tous les algorithmes ne mettent pas le même temps pour réaliser une même tâche sur les mêmes données. Cette différence n'est pas très claire lorsque la liste à trier comporte moins de dix éléments environ. Au contraire, lorsque la liste à trier comporte des centaines de milliers d'éléments, ce qui est courant en informatique, la différence devient énorme. Bien entendu, le nombre d'opérations effectuées par l'algorithme (et donc le temps de calcul) dépend de la longueur de la liste.

De plus, pour un algorithme donné, le nombre précis d'opérations effectuées dépend souvent des données exactes sur lesquelles le calcul est effectué, c'est pourquoi, en algorithmique, les notions de complexités importantes sont la complexité « dans le pire des cas », ou la complexité « en moyenne », dans le but de lisser ces variations. Ainsi, le nombre d'opérations nécessaires est vu comme une fonction de la seule longueur de la liste à trier. Enfin, ce nombre précis d'opérations dépend aussi des détails de l'implémentation du programme, et c'est pourquoi ce qui est important, c'est l'ordre de grandeur de cette fonction, plutôt que ses valeurs (le plus souvent la notation grand  $O(\dots)$  de Landau est utilisée).

Plus précisément, sur une liste de  $n$  éléments, le tri par minimum, le tri par insertion ou le tri à bulles utiliseront un nombre d'opérations de l'ordre de  $n^2$  en moyenne, alors que le tri rapide et le tri fusion utiliseront de l'ordre de  $n \times \log(n)$  opérations. Ces derniers sont donc plus efficaces. Lorsque le tri est effectué en classe entière, la différence est perceptible par les élèves. Cette première approche de la complexité des algorithmes montre qu'il est important d'utiliser des algorithmes efficaces pour obtenir des programmes rapides.



---

 TRIER POUR  
 MIEUX COMPARER
 

---

Les calculs permettant l'analyse « en moyenne » sont souvent relativement compliqués, puisqu'ils nécessitent de passer par un modèle probabiliste des instances du problème. La complexité « dans le pire des cas » est souvent plus facile à déterminer.

L'un des cas les plus simples est celui du tri à bulles : pour trier une liste de  $n$  nombres avec cet algorithme, combien faut-il *au maximum* de comparaisons et d'échanges ?

Lors du premier parcours, il s'agit de comparer la valeur de l'élément 1 avec celle de l'élément 2, puis la valeur de l'élément 2 avec celle de l'élément 3, ... et enfin, la valeur de l'élément  $n - 1$  avec celle de l'élément  $n$ . Au total, il y a ici  $n - 1$  comparaisons. Dans le pire des cas (lorsque le plus grand élément de la liste était en position 1), ce parcours induit également  $n - 1$  échanges. À la fin du premier parcours, le maximum de la liste a atteint sa place définitive, et il n'est plus nécessaire de s'en occuper. Ainsi, lors du second parcours, le nombre de comparaisons n'est plus que de  $n - 2$ , et dans le pire des cas, le nombre d'échanges est aussi de  $n - 2$ . À chaque parcours, le nombre de comparaisons diminue de 1, et dans le pire des cas, le nombre d'échanges est égal au nombre de comparaisons. Maintenant, quel est le nombre de parcours ? L'algorithme s'arrête lorsqu'il n'y a plus aucun échange lors d'un parcours. Le pire des cas est celui où, au départ, la liste de nombres était triée dans l'ordre décroissant : dans ce cas, le nombre de parcours nécessaires est  $n - 1$ .

Au maximum, le nombre de comparaisons (et d'échanges) pour le tri à bulles est donc :

$$(n - 1) + (n - 2) + \dots + 2 + 1 = \frac{n(n - 1)}{2},$$

c'est-à-dire au final  $n(n - 1)$  opérations dans le pire des cas. En ne s'intéressant qu'à l'ordre de

grandeur de cette fonction, le  $O(n^2)$  annoncé ci-dessus, pour la complexité en moyenne du tri à bulles, est retrouvé.

Il est aussi possible de se demander quel est l'algorithme de tri le plus rapide de tous. Les chercheurs en algorithmique ont démontré que tout algorithme de tri généraliste basé sur des comparaisons, comme ceux présentés ici, utilise au moins de l'ordre de  $n \times \log(n)$  opérations pour trier une liste de  $n$  éléments dans le pire des cas. Le tri fusion ou le tri rapide font donc partie des algorithmes optimaux dans leur catégorie.

Cependant, il existe des algorithmes de tri spécialisés, fonctionnant sur d'autres principes, qui sont plus efficaces. Par exemple, s'il est connu à l'avance que les nombres à trier sont tous des nombres à un chiffre (entre 0 et 9), il suffit de parcourir la liste en comptant le nombre de 0, le nombre de 1, le nombre de 2, etc., puis de créer une liste triée en écrivant d'abord tous les 0, puis tous les 1, puis tous les 2, etc. Le nombre d'opérations effectuées en exécutant cet algorithme de « tri par comptage » pour trier une liste de  $n$  chiffres est de l'ordre de  $n$ .

Cette mesure de complexité ne prend en compte que le temps de calcul, via le nombre d'opérations effectuées pendant l'exécution des algorithmes. Il est toutefois également intéressant de se préoccuper de l'espace mémoire utilisé par ces algorithmes, en plus du temps nécessaire pour les exécuter. C'est une question que les chercheurs en informatique se sont posée, et ils ont retrouvé des problématiques proches de celles concernant le temps de calcul. En ce qui concerne les algorithmes de tri, une question pertinente dans ce cadre est de se demander si l'algorithme peut être programmé de façon à être exécuté « en place », c'est-à-dire sans utiliser une deuxième liste

auxiliaire de même longueur que la liste initiale. C'est assez facile pour le tri insertion, le tri par minimum, le tri à bulles et le tri rapide, mais c'est beaucoup plus difficile, quoique pas impossible, à réaliser pour le tri fusion.

De plus, avec la construction des ordinateurs multi-cœurs (avec plusieurs processeurs, plusieurs chefs d'orchestre), les questions de complexité en temps et mémoire sont à revisiter, et certains algorithmes comme le tri fusion se prêtent plus naturellement à la parallélisation que d'autres algorithmes de tri.

Enfin, la diversité des algorithmes de tri permet d'illustrer de nombreux types d'algorithmes. Par exemple, certains d'entre eux, comme le tri fusion ou le tri rapide se prêtent à une programmation récursive, un concept clé des langages de programmation dits fonctionnels, comme Caml. Ce paradigme de programmation permet de faire plus simplement des preuves mathématiques par récurrence de la correction des programmes écrits, c'est-à-dire prouver que le programme trie bien la liste donnée en entrée.

## 5. — Conclusion

Trier des données, une activité omniprésente en informatique, nécessite de savoir les comparer deux à deux. Quant à la notion de comparaison de nombres, c'est un incontournable des mathématiques. En conciliant ces deux aspects, l'activité d'initiation aux algorithmes de tri proposée permet aux élèves de les travailler simultanément. Cette activité relève de l'Informatique Sans Ordinateur : ce sont les élèves qui jouent le rôle des données à trier, mais aussi le rôle du processeur exécutant l'algorithme de tri, de manière ludique et collaborative.

Le travail mené autour de cette activité a été l'occasion, d'une part, de faire le lien entre informatique et mathématiques, et de rendre plus légitimes les tâches de comparaison de nombres, et d'autre part, de réfléchir à l'intégration de l'enseignement de concepts informatiques dans les progressions mathématiques. C'est pourquoi nous continuons à mener ce travail de réflexion, pour insérer d'autres activités du groupe « Faire de l'informatique sans ordinateur » dans les progressions de mathématiques.

## Références

- [1] Knuth, D. E., *The Art of Computer Programming*, 3, Sorting and Searching, Addison Wesley (1973).
- [2] Hoare, C. A. R., "Quicksort," *Computer Journal* 5, pp. 10-15 (1962).