Algorithmique : le tri à bulles

Fiche professeur

2e

Auteur : Jean-Marc Duquesnoy et Raymond Moché

But de l'activité : Activité d'initiation à l'algorithmique. Programmer et tester le tri à bulles permet d'aborder simplement les instructions d'affectation et les boucles « pour » et « si... alors fin ». L'essentiel de cette activité se fait en langage naturel. Rappelons que des tris interviennent obligatoirement dans les calculs de médiane, quartiles, déciles et boîtes à moustaches.

Compétences engagées :
- ✓ Gestion des entrées et sorties.
- ✓ Instructions d'affectation.
- ✓ Boucle pour.
- ✓ Boucle si ... alors ... fin.
- ✓ Comprendre un algorithme en langage naturel.
- ✓ Tester un algorithme donné sur « Scilab pour les lycées » ou « Xcas ».
- ✓ Comparer les vitesses d'exécution de divers algorithmes.
- ✓ Simulation d'entiers au hasard entre 2 bornes
- ✓ Simulation de réels au hasard sur un intervalle

Pré-requis :
- ✓ La simulation de nombres au hasard a été rencontrée en 3ème.

Matériels utilisés :
- ✓ Salle informatique (de préférence) ou salle de classe
- ✓ Un vidéo-projecteur et un ordinateur muni de « Scilab pour les lycées » ou « Xcas » pour le professeur, si l'activité est faite en classe.

Durée indicative : 2 heures

Noms des logiciels utilisés : « Scilab pour les lycées » ou « Xcas ».

Documents utiles à télécharger :
- ✓ « Fiche Élève ».
- ✓ « TriBulles.sci » (fichier scilab).
- ✓ « TriBullesXcas.xws« (fichier Xcas).

Déroulement de la séance :

Les 4 premières questions (sur 5) concernent d'élaboration de l'algorithme du « tri à bulles » en langage naturel. Elles peuvent se faire en classe. Une bonne partie de ce travail consistera à comprendre l'algorithme, qui est décrit soigneusement (?). Il serait préférable que la 5ème question soit faite individuellement par les élèves devant un ordinateur. Mais on peut aussi imaginer que le professeur manipule lui-même et montre ses calculs à ses élèves à l'aide d'un vidéo-projecteur. Dans ce cas, il serait utile de leur distribuer des photocopies de « TriBulles.sci » ou de « TriBullesXcas.xws » suivant le cas. Ces algorithmes exécutables sont donnés parce que cette activité est une activité d'initiation. Traduire l'algorithme **4** en un algorithme exécutable est un exercice plutôt difficile, surtout avec « Xcas » qui demande un investissement plus important. Les élèves devront enfin les tester dans divers cas. Ils constateront qu'utiliser des fonctions pré-programmées est beaucoup plus rapide. Il y a plusieurs raisons :

- d'abord, on pourrait optimiser « TriBulles.sci » ou « TriBullesXcas.xws » (par exemple, il est inutile de ré-éxaminer les couples qui sont dans leur position définitive). L'algorithme optimisé resterait médiocre (voir l'article « Tri à bulles » de Wikipedia).

- il y a des algorithmes dont la conception est meilleure ;

- « scilab » ou « Xcas » sont des langages interprétés. Il y a des langages plus efficaces.

Solution, autres commentaires :

1 - l'algorithme 1 produit b, b

2 -

$$\boxed{\begin{array}{l} c \leftarrow a\,; \\ a \leftarrow b\,; \\ b \leftarrow c\,; \end{array}}$$

3 - Il en faut 4 : c'est le cas le plus défavorable. Au contraire, pour ranger dans l'ordre croissant $1, 2, 3, 4, 5$, il n'y a rien à faire, mais l'algorithme imposera quand même 4 manipulations de base.

4 - Il suffit d'ajouter l'instruction « Aoriginal=A ; » juste après début et donc avant que ne commencent les transformations de A. Ensuite, remplacer « Afficher A » par « Afficher $Aoriginal, A$ ».

5 - Voir les commentaires dans « TriBulles.sci » ou « TriBullesXcas.xws ». Les générateurs de nombres

aléatoires sont indiqués.

Les élèves ont dû rencontrer des simulations de nombres au hasard en 3ème (fonctions ALEA ou ALEA.ENTRE.BORNES du tableur Calc d'OOo). Ceci dit, le Calcul des probabilités intervient ici pour l'unique raison qu'il permet d'écrire instantanément de longues listes de nombres, dont la signification ne nous intéresse pas et qui servent à tester les algorithmes. On n'imagine pas saisir une liste de 100000 nombres à la main !

Bien entendu, il est important de manipuler d'aussi longues listes pour convaincre les élèves que le calcul automatique est indispensable dans les applications.

Les fonctions de tri « trier » ou « SortD » ordonnant dans l'ordre décroissant, il faut faire une petite astuce : comme

$$x = 1 - (1 - x),$$

en ordonnant dans l'ordre décroissant les valeurs de $1 - x$, on récupère au premier membre les valeurs de x rangées dans l'ordre croissant.