

Algorithmique

Gilles Aldon, Jérôme Germoni, Jean-Manuel Mény

IREM de Lyon

Mars 2012

Plan prévisionnel du stage

- 1 Introduction
 - 1 L'algorithmique dans les programmes du lycée
 - 2 Algorithme / programme
 - 3 « Axes d'analyse »
- 2 Premiers algorithmes et programmes
- 3 Complexité d'un algorithme
- 4 Arbres et forêts (cachés)
- 5 Paradigmes de programmation (E. Coquery)
- 6 Algorithmes de tri
- 7 Algorithmes gloutons

Programme (seconde et au-delà)

Programme (seconde et au-delà)

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique.

Au collège, les élèves ont rencontré des algorithmes (algorithmes opératoires, algorithme d'Euclide, algorithmes de construction en géométrie).

Programme (seconde et au-delà)

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique.

Au collège, les élèves ont rencontré des algorithmes (algorithmes opératoires, algorithme d'Euclide, algorithmes de construction en géométrie).

Ce qui est proposé dans le programme est une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel.

Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul.

Programme de la classe de seconde

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) .

Informatique et sciences du numérique (ISN)

Quatre axes dans le projet de programme :

- La représentation de l'information
- L'algorithmique
- Langages et programmation
- L'architecture matérielle

Algorithme et programmation

Algorithmique : langage naturel

Programmation : langage spécifique, syntaxe spécifique (voire machine spécifique)

Au lycée, apprentissage conjoint

Algorithmique et programmation

- Algorithmique : souplesse du langage naturel, signification explicite (présentation des objectifs de certaines étapes, précisions techniques par étapes)

Algorithmique et programmation

- Algorithmique : souplesse du langage naturel, signification explicite (présentation des objectifs de certaines étapes, précisions techniques par étapes)
- Programmation : contraintes fortes, la syntaxe obscurcit souvent la signification

Algorithmique et programmation

- Algorithmique : souplesse du langage naturel, signification explicite (présentation des objectifs de certaines étapes, précisions techniques par étapes)
- Programmation : contraintes fortes, la syntaxe obscurcit souvent la signification
- mais on peut rapprocher un texte de programme d'un texte d'algorithme par ajout de commentaires.

Algorithmique et programmation

- Algorithmique : contraintes floues. Qu'a-t-on le droit de faire ? Quelles sont les contraintes ? Difficulté de compréhension sur la nature de l'algorithmique dans la phase de découverte.
- Programmation : contraintes fortes, clarifie ce qu'on a le « droit » de faire (une exploration des menus algobox donne par exemple les outils autorisés), sanction de la machine si on sort des possibles et/ou de la logique.

La programmation est nécessaire à la construction de l'idée d'algorithme.

Algorithmique et programmation

- Algorithmique : contraintes floues. Qu'a-t-on le droit de faire ? Quelles sont les contraintes ? Difficulté de compréhension sur la nature de l'algorithmique dans la phase de découverte.
- Programmation : contraintes fortes, clarifie ce qu'on a le « droit » de faire (une exploration des menus algobox donne par exemple les outils autorisés), sanction de la machine si on sort des possibles et/ou de la logique.
- Faire tourner un programme : observations, réactions. . .

La programmation est nécessaire à la construction de l'idée d'algorithme.

Algorithme et programmation : nature des objets en jeu

Un exemple utilisé en seconde.

- Entrées : les coordonnées de deux points A et B.
- Sortie : les coordonnées du milieu du segment $[AB]$.
- Traitement : calculer la moyenne des abscisses, la moyenne des ordonnées.

Algorithme et programmation : nature des objets en jeu

Un exemple utilisé en seconde.

- Entrées : les coordonnées de deux points A et B.
- Sortie : les coordonnées du milieu du segment $[AB]$.
- Traitement : calculer la moyenne des abscisses, la moyenne des ordonnées.
 - Sur une calculatrice Ti 82 par exemple, le texte du programme est celui de l'algorithme (à la syntaxe près).
 - Mais avec les entrées $A(\sqrt{2}; 1)$, $B(1; 1)$, on n'obtient pas les coordonnées du milieu.
 - L'algorithme calcule les coordonnées du milieu
 - mais pas sa traduction directe en machine.
 - À rapprocher de objet géométrique / dessin ?

Algorithme et programmation

L'algorithme est un outil théorique :

- il se prouve (validité, correction),
- il **prouve**.

En revanche, ses traductions en machine (programmes) ne prouvent pas.

Différentes questions sur un algorithme

- Déroulement de l'algorithme
 - terminaison
 - correction
 - complexité
- Types d'algorithmes
 - glouton
 - diviser pour régner
 - *backtracking*
- structure des données
 - types basiques : nombres, listes
 - plus de structure : arbres, graphes
 - codage des structure
- paradigmes de programmation