

Algorithmique¹

L'algorithmique est l'étude et la production de règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution d'un problème permettant de décrire précisément des étapes pour résoudre un problème algorithmique. En d'autres termes, un algorithme est une suite finie et non-ambiguë d'instructions permettant de donner la réponse à un problème.

¹ [Contenu soumis à la licence CC-BY-SA 3.0](#). Source : Article [Algorithmique](#) de [Wikipédia en français](#) ([auteurs](#)).

Table des matières

1	Étymologie.....	1
2	Histoire.....	1
2.1	Antiquité.....	1
2.2	Étude systématique.....	1
2.3	L'époque contemporaine.....	2
3	Vocabulaire.....	2
4	Étude formelle.....	3
4.1	Structures algorithmiques.....	3
4.2	Correction, complétude, terminaison.....	3

Index lexical

complétude.....	3 sv
machines de Turing.....	2
terminaison.....	3

1 Étymologie

Le mot « algorithme » vient du nom du mathématicien Al-Khwarizmi² (latinisé au Moyen Âge en Algoritmi), qui, au IX^e siècle écrit le premier ouvrage systématique donnant des solutions aux équations linéaires et quadratiques. « Algorithmes » a donné « algorithmique » auxquels certains préfèrent le néologisme « algorithmie ».

2 Histoire

2.1 Antiquité

Les premiers algorithmes dont on a retrouvé des descriptions datent des Babyloniens, au III^e millénaire av. J.-C.. Ils décrivent des méthodes de calcul et des résolutions d'équations à l'aide d'exemples.

Un algorithme célèbre est celui qui se trouve dans le livre 7 des *Éléments* d'Euclide, et appelé algorithme d'Euclide. Il permet de trouver le plus grand diviseur commun, ou PGCD, de deux nombres. Un point particulièrement remarquable est qu'il contient explicitement une itération et que les propositions 1 et 2 démontrent sa correction.

C'est Archimède qui proposa le premier un algorithme pour le calcul de π .

2.2 Étude systématique

Le premier à avoir systématisé des algorithmes est le mathématicien arabophone Al Khuwarizmi, actif entre 813 et 833. Dans son ouvrage *Abrégé du calcul par la restauration et la comparaison*, il étudie toutes les équations du second degré et en donne la résolution par des algorithmes généraux. Il utilise des méthodes semblables à celles des Babyloniens, mais se différencie par ses explications systématiques là où les Babyloniens donnaient seulement des exemples.

Le savant arabe Averroès (1126-1198) évoque une méthode de raisonnement où la thèse s'affine étape par étape, itérativement, jusqu'à une certaine convergence et ceci conformément au déroulement d'un algorithme. À la même époque, au XII^e siècle, le moine Adelard de Bath introduit le terme latin de algorismus, par référence au nom de Al Khuwarizmi. Ce mot donne algorithme en français en 1554.

Au XVII^e siècle, on pourrait entrevoir une certaine allusion à la méthode algorithmique chez René Descartes dans la méthode générale proposée par le *Discours de la mé-*

² Il ne faut pas confondre ce mathématicien avec un autre auteur perse : Abu-'Abdollah Mohammad Khuwârizmi qui, lui, est l'auteur de *Mafâtih al-'Olum* (*Les clés des sciences*), encyclopédie écrite dans les années 976-997.

thode (1637), notamment quand, en sa deuxième partie, le mathématicien français propose de :

« *diviser chacune des difficultés que j'examinerois, en autant de parcelles qu'il se pourroit, et qu'il seroit requis pour les mieux résoudre.* »

Sans évoquer explicitement les concepts de boucle, d'itération ou de dichotomie, l'approche de Descartes prédispose la logique à accueillir le concept de programme, mot qui naît en français en 1677.

L'utilisation du terme algorithme est remarquable chez Ada Lovelace (voir l'illustration page 4), fille de Lord Byron et assistante de Charles Babbage (1791-1871).

2.3 L'époque contemporaine

L'algorithmique du XXe et du XXIe siècle se base souvent sur des formalismes comme celui des machines de Turing, qui permettent de définir précisément ce qu'on entend par « étapes », par « précis » et par « non ambigu » et elle donne un cadre scientifique pour étudier les propriétés des algorithmes. Cependant, le type de formalisme choisi engendre des algorithmes différents pour résoudre un même problème, par exemple l'algorithmique récursive, l'algorithmique parallèle ou l'informatique quantique donnent lieu à des algorithmes différents de ceux de l'algorithmique itérative issue de la machine de Turing.

Avec l'informatique, l'algorithmique s'est beaucoup développée dans la deuxième moitié du XX^e siècle et Donald Knuth, auteur du traité *The Art of Computer Programming*, qui décrit de très nombreux algorithmes, en a posé des fondements mathématiques rigoureux de leur analyse.

3 Vocabulaire

Le substantif algorithmique désigne la méthode utilisant des algorithmes. Le terme est également employé comme adjectif.

Un algorithme énonce une résolution sous la forme d'une série d'opérations à effectuer. La mise en œuvre de l'algorithme consiste en l'écriture de ces opérations dans un langage de programmation et constitue alors la brique de base d'un programme informatique.

Les informaticiens utilisent fréquemment l'anglicisme implémentation pour désigner cette mise en œuvre. L'écriture en langage informatique est aussi fréquemment désignée par le terme « codage », qui n'a ici aucun rapport avec la cryptographie, mais qui se réfère au terme « code source » pour désigner le texte, en langage de programmation, constituant le programme. L'algorithme devra être plus ou moins détaillé selon le niveau

d'abstraction du langage utilisé, de même qu'une recette de cuisine doit être plus ou moins détaillée selon l'expérience du cuisinier.

4 Étude formelle

De nombreux outils formels ou théoriques ont été développés pour décrire les algorithmes, les étudier, exprimer leurs qualités, pouvoir les comparer :

- ainsi, pour décrire les algorithmes, des structures algorithmiques ont été mises en évidence : structures de contrôle et structures de données ;
- pour justifier de la qualité des algorithmes, les notions de correction, de complétude et de terminaison ont été mises en place ;
- enfin, pour comparer les algorithmes, une théorie de la complexité des algorithmes a été définie.

4.1 Structures algorithmiques

Les concepts en œuvre en algorithmique, par exemple selon l'approche de N. Wirth pour les langages les plus répandus (Pascal, C, etc.), sont en petit nombre. Ils appartiennent à deux classes :

- les structures de contrôle :
 - séquences,
 - conditionnelles,
 - boucles ;
- les structures de données :
 - constantes,
 - variables,
 - tableaux,
 - structures récursives (listes, arbres, graphes).

Ce découpage est parfois difficile à percevoir pour certains langages (Lisp, Prolog...) plus basés sur la notion de récursivité où certaines structures de contrôle sont implicites et, donc, semblent disparaître.

4.2 Correction, complétude, terminaison

Ces trois notions « correction », « complétude », « terminaison » sont liées, et supposent qu'un algorithme est écrit pour résoudre un problème.

La terminaison est l'assurance que l'algorithme terminera en un temps fini. Les preuves de terminaison font habituellement intervenir une fonction entière positive strictement décroissante à chaque « pas » de l'algorithme.

Étant donnée la garantie qu'un algorithme terminera, la preuve de correction doit apporter l'assurance que si l'algorithme termine en donnant une proposition de solution, alors cette solution est correcte — c'est-à-dire qu'elle est effectivement une solution au problème posé.

La preuve de complétude garantit que, pour un espace de problèmes donné, l'algorithme, s'il termine, donnera une solution pour chacune des entrées.



Illustration 1: Ada Lovelace