Une introduction à la science informatique

pour les enseignants de la discipline en lycée

Collection Repères pour agir
série Disciplines et Compétences

Coordination : Centre régional de documentation pédagogique de l'académie de Créteil
Directrice de collection : Christine Moulin, IA-IPR d'allemand
Responsable éditorial : Gilles Gony
Couverture : Dominique Florentin

Une introduction à la science informatique
pour les enseignants de la discipline en lycée

Édité par le Centre régional de documentation pédagogique de l'académie de Paris
Directrice : Marie-Christine Ferrandon
Chargé de mission Édition-TICE : Michel Bézard
Révision : Laure Técher
Édité avec le soutien de l'Association des sciences et techniques de l'information, et celui d'Enseignement public et informatique

Impression : Jouve
ISBN : 978-2-86631-188-9
ISNN : 1625-3000

Une introduction à la science informatique

pour les enseignants de la discipline en lycée

dirigé par Gilles Dowek

préface de Gérard Berry,
professeur au Collège de France

Auteurs

Préface : Gérard Berry
Professeur au Collège de France
Direction de l'ouvrage : Gilles Dowek
Directeur de recherche à l'Institut national de recherche en informatique et en automatique

Jean-Pierre Archambault
Chargé de mission au CNDP-CRDP Paris
Emmanuel Baccelli
Chargé de recherche à l'Institut national de recherche en informatique et en automatique
Sylvie Boldo
Chargé de recherche à l'Institut national de recherche en informatique et en automatique
Denis Bouhineau
Maître de conférences à l'université Joseph-Fourier, Grenoble
Patrick Cégielski
Professeur à l'université Paris-Est Créteil
Thomas Clausen
Maître de conférences à l'École polytechnique
Irène Guessarian
Professeur émérite à l'université Pierre-et-Marie-Curie, chercheur au Laboratoire d'informatique algorithmique : Fondements et Applications
Stéphane Lopès
Maître de conférences à l'université de Versailles Saint-Quentin
Laurent Mounier
Maître de conférences à l'université Joseph-Fourier, Grenoble
Benjamin Nguyen
Maître de conférences à l'université de Versailles Saint-Quentin
Franck Quessette
Maître de conférences à l'université de Versailles Saint-Quentin
Anne Rasse
Maître de conférences à l'université Joseph-Fourier, Grenoble
Brigitte Rozoy
Professeur à l'université de Paris-Sud
Claude Timsit
Professeur à l'université de Versailles Saint-Quentin
Thierry Viéville
Directeur de recherche à l'Institut national de recherche en informatique et en automatique
Jean-Marc Vincent
Maître de conférences à l'université Joseph-Fourier, Grenoble

Les auteurs remercient Nicolas Boullis, Julien Cervelle, Jean-Paul Delahaye, Ahmed Djebbar, Lena Domröse, Philippe Jacquet, Dominique Lacroix, Bernard Lang, Jérémie Lumbroso, Maryse Pelletier-Koskas et Tuong Vinh pour leur aide pendant la rédaction de cet ouvrage.

Ils souhaitent remercier également Gérard Berry, Robert Cabane, Pascal Guitton et Maurice Nivat sans qui cette entreprise n'aurait jamais été possible.

Sommaire

Préface

Gérard Berry
Professeur au Collège de France

C'est un grand plaisir que de préfacer cet ouvrage collectif, qui marque l'entrée de la science informatique en tant que discipline autonome dans l'enseignement secondaire. Depuis sa naissance dans les années 1950, et grâce à sa très grande flexibilité, l'informatique a successivement transformé tous les pans de la société : industrie, communication, commerce, culture, etc. Son essor s'accentue encore avec l'explosion d'Internet et la généralisation de l'informatisation des objets. Si elle reste souvent mystérieuse et quelquefois hostile pour les adultes, elle fait partie intégrante du monde des enfants d'aujourd'hui, au même titre que l'eau courante ou l'électricité. Il faut donc cesser de l'appeler une « nouvelle technologie » : si cette expression a encore un sens pour les adultes du XX^e^ siècle, elle n'en aura jamais pour les enfants du XXI^e^ siècle, ceux à qui l'enseignement secondaire s'adresse. Ils ne pourront comprendre l'avant-informatique que par les cours d'histoire.

Dans l'éducation secondaire française, l'informatique a longtemps été réduite à une technique auxiliaire dont il fallait simplement s'approprier l'usage. C'était négliger deux points fondamentaux. D'abord, son rôle de ferment d'une créativité extraordinaire, qui s'exprime dans de multiples applications non prévues encore récemment mais à l'essor foudroyant : diffusion à grande échelle de musique ou de films, moteurs de recherche permettant l'exploitation de masses gigantesques de données, réseaux sociaux créant des formes de communication qui s'affranchissent de l'espace et du temps, etc. Les résultats de cette créativité permanente modifient nos façons de faire les plus courantes à un point que nous ne réalisons peut-être pas encore complètement. Ils sont aussi associés à un essor non moins grand en nombre d'emplois qualifiés et intéressants dans le monde entier. Ensuite, le fait que cette créativité n'est pas seulement permise par l'avancée technologique. Elle a comme fondement les avancées constantes de la science informatique, qui est au cœur des machines, réseaux et logiciels modernes autant la science physique est au cœur des machines mécaniques, électriques ou électroniques. Après plus de soixante ans d'existence, cette science originale et féconde ne peut plus être appelée récente. Elle occupe des centaines de milliers

de chercheurs et d'enseignants dans le monde. Elle est assise sur un corpus théorique considérable mais aussi de nature profondément expérimentale. Elle conduit à de nouvelles connaissances et à de nouvelles façons de voir et de faire. Contrairement aux sciences de la nature, elle n'a pas comme préalable l'étude nécessairement lente et délicate de phénomènes complexes indépendants de notre volonté. Au contraire, elle construit ce qu'elle étudie avec très peu de barrières, ce qui explique sa vitesse d'expansion. De plus, les concepts et méthodes informatiques pénètrent maintenant toutes les sciences de la nature, même celles qui étaient traditionnellement peu touchées par les formalismes et méthodes mathématiques comme la biologie ou certaines sciences sociales, en leur fournissant des façons radicalement nouvelles de modéliser et de comprendre les phénomènes.

Il faut malheureusement constater que, malgré la qualité de sa recherche, notre pays n'est toujours pas un grand acteur de ce bouillonnement créateur, et se pose surtout en utilisateur et simple consommateur de technologies informatiques made in USA or Asia. *Pour ne pas continuer à accumuler le retard sur ces évolutions, il est indispensable que la science informatique et ses conséquences soient davantage connues dans tous les pans du tissu social, ce qui impose de lui donner sa place dans les cursus scolaires. C'est ce qui va heureusement recommencer en 2012 avec l'introduction de l'informatique en tant qu'enseignement de spécialité en terminale. Mais, avant de former les élèves, il faut former les professeurs. C'est l'objectif de ce livre, qui prend d'emblée un point de vue résolument moderne en abordant directement le cœur du sujet : l'informatique est fondée sur l'interaction et l'équilibre de quatre notions fondamentales, le codage numérique de l'information, les algorithmes ou procédés de calcul automatiques, les langages permettant de définir formellement les algorithmes, et les machines exécutant les programmes écrits dans ces langages. Ces notions sont bien sûr interdépendantes, mais elles sont suffisamment autonomes pour être décrites en chapitres bien articulés. Le livre s'intéresse aussi à de grands domaines d'applications conceptuellement formateurs et essentiels pour les applications en pratiques : l'architecture de machines, celle de réseaux et les bases de données. Les différents chapitres ne font pas que présenter brutalement les notions. Ils proposent des exercices, discutent des questions pédagogiques, et suggèrent des voies d'approfondissement.*

Ce travail sera certainement destiné à évoluer en fonction des retours des professeurs et des élèves, de l'évolution à venir des objectifs et programmes d'enseignement, et de celle du sujet lui-même. Et il sera bien sûr complété par d'autres contributions internes ou externes dans la myriade des formes proposées par les outils de documentation et communication modernes. L'informatique n'est-elle elle-même pas un merveilleux sujet pour développer des contenus créatifs sous toutes les formes, textuelles, audiovisuelles et interactives ?

Introduction

Depuis plus de quatre mille ans, nous connaissons des algorithmes qui permettent d'effectuer des opérations arithmétiques, calculer des intérêts composés, déterminer l'aire de surfaces agricoles, dériver des expressions fonctionnelles, résoudre des systèmes d'équations linéaires, fabriquer des tissus, préparer des aliments… Le mot « algorithme » lui-même dérive du nom du mathématicien al-Khwarizmi, auteur, au IXe siècle, d'un *Livre de l'addition et de la soustraction d'après le calcul indien*, qui systématise les opérations arithmétiques en numération décimale à position. Exécuter ces algorithmes ne demandant que d'effectuer, systématiquement et sans réfléchir, une suite d'opérations, l'idée est vite apparue de s'aider d'outils pour le faire. Ainsi, les baguettes à calculer, les bouliers, les échiquiers, les métiers à tisser, les machines mécanographiques… nous apparaissent rétrospectivement comme les précurseurs de nos ordinateurs.

Algorithmes et machines

Une révolution s'est cependant produite au milieu du XXe siècle, quand la technique des tubes à vide a permis de réaliser les premiers ordinateurs, c'est-à-dire les premières machines à calculer universelles : depuis longtemps, nous avions construit des machines capables d'exécuter plusieurs algorithmes, comme la machine de Pascal, qui effectuait des additions et des soustractions, la nouveauté avec les ordinateurs était qu'ils étaient capables d'exécuter, non seulement plusieurs algorithmes, mais tous les algorithmes possibles, opérant sur des données symboliques. C'est cette universalité des ordinateurs qui explique leur omniprésence dans nos vies professionnelles et personnelles : un comptable, un architecte et un médecin n'utilisent pas les mêmes algorithmes, mais ils utilisent tous un ordinateur pour les exécuter.

L'histoire de la naissance de l'informatique est donc celle de la rencontre de ces deux concepts d'algorithme et de machine. Un *algorithme* est une méthode opérationnelle qui permet de résoudre systématiquement toutes les instances d'un problème donné. Une *machine* est un système physique, avec lequel nous avons défini un protocole d'échange d'informations. Ces deux concepts sont très vastes : par exemple, une bille que nous laissons tomber dans le vide pen-

dant un temps donné et dont nous mesurons la distance parcourue est une machine qui calcule le carré d'un nombre – en choisissant judicieusement les unités de temps et de distance. Les machines ne se limitent donc pas aux ordinateurs, tels que celui qui se trouve dans notre cartable.

Langages et informations

Dans les toutes premières années de l'informatique, deux nouveaux concepts sont venus s'ajouter à ces deux concepts structurants d'algorithme et de machine. Tout d'abord est apparue la nécessité de décrire les algorithmes que nous souhaitons voir exécutés dans un *langage* compréhensible à la fois par la machine qui exécute l'algorithme et par l'être humain qui le décrit. C'est ainsi que sont apparus les premiers langages de programmation et, avec eux, les premiers programmes, incarnations d'un algorithme dans un langage particulier. Les langages de programmation ont une certaine parenté avec les langues naturelles que nous utilisons tous les jours ; on y distingue, par exemple, des *expressions* qui sont les homologues de nos groupes nominaux et des *instructions*, exprimées par un verbe à l'impératif, qui sont les homologues de nos phrases. Mais ils ont surtout de nombreux points communs avec d'autres langages formels, utilisés avant eux : la notation musicale, la notation algébrique, la nomenclature chimique, etc. Avec la naissance de l'informatique, nous avons commencé à nous poser quotidiennement des questions, rares jusqu'alors : quel langage inventer pour décrire tels ou tels objets ? Cette interrogation sur la notion de langage place l'informatique dans le prolongement de la linguistique, mais surtout de la logique.

Le second concept est celui d'*information*. Un ordinateur, nous l'avons vu, ne peut exécuter que des algorithmes opérant sur des données symboliques, c'est-à-dire des données qui, tels des nombres entiers ou des textes, peuvent s'exprimer par des suites finies de symboles, chiffres ou lettres, pris dans un ensemble fini. De telles données symboliques s'appellent des *informations*. Comme nous le verrons, l'ensemble de symboles choisis importe peu et toutes les informations peuvent s'exprimer avec deux symboles seulement, conventionnellement écrits 0 et 1.

Les images et les sons ne sont pas des données symboliques. Toutefois, ils peuvent être numérisés, c'est-à-dire représentés sous la forme de telles données. Bien entendu, nous perdons un peu de qualité en numérisant une image ou un son, mais parfois suffisamment peu pour que notre œil ou notre oreille ne s'en aperçoivent pas, *grosso modo* parce que ces organes, eux aussi, numérisent les images et les sons. Représenter les nombres, les textes, les images et les sons de manière uniforme, avec des 0 et des 1, permet de les transmettre

sur un réseau, de les conserver sur un disque, etc. en utilisant des méthodes uniformes, alors que, par le passé, nous utilisions des méthodes différentes – vinyles, pellicules, etc. – pour conserver, par exemple, les sons et les images. Avec la notion d'information apparaissent aussi une notion de quantité d'information et différentes méthodes de quantification – la quantité d'information au sens de Kolmogorov ou de Shannon –, selon que l'on s'intéresse à l'information contenue dans un message ou à l'information apportée par un message.

Quatre concepts indissociables

L'informatique nous apparaît donc comme structurée par les quatre concepts d'algorithme, de machine, de langage et d'information. Chacun de ces concepts est bien entendu antérieur à l'informatique : la nouveauté est dans l'articulation de ces concepts issus d'univers scientifiques et techniques si différents. Au début du XX^e^ siècle, les spécialistes de la notion d'algorithme étaient les mathématiciens, les comptables, etc., les spécialistes de la notion de machine étaient les ingénieurs qui les fabriquaient et les physiciens qui en étudiaient les principes – au sens où la thermodynamique étudie les principes de la machine à vapeur –, les spécialistes de la notion de langage étaient les linguistes, les logiciens, les traducteurs, etc. et les spécialistes de la notion d'information, les bibliothécaires, les imprimeurs, les agents du chiffre, etc. On aurait été alors très surpris d'apprendre que, quelques années plus tard, ces professionnels issus d'horizons si variés se mettraient à coopérer pour créer une science nouvelle et, surtout, que cette science atteindrait un tel degré de cohérence : pour concevoir un langage qui permet de décrire des algorithmes opérant sur des informations, afin qu'ils soient exécutés par une machine, il ne suffit pas d'être expert dans l'une de ces notions, il faut avoir une connaissance relativement approfondie de chacune d'elles.

Les évolutions récentes de l'informatique illustrent l'indissociabilité de ces quatre concepts. Les machines, nous l'avons vu, sont d'une grande diversité : un réseau qui relie des ordinateurs entre eux est une machine et une mémoire de masse qui permet de stocker de grandes quantités d'informations également. Ces deux types de machines ont permis l'apparition de nouveaux usages, qui constituent une évolution importante de l'informatique : pour nombre de leurs utilisateurs, les ordinateurs ne servent en effet pas à transformer des informations en leur appliquant un algorithme, mais plus simplement à transmettre ces informations d'un bout à l'autre d'un réseau et à les stocker pour les retrouver plus tard, c'est-à-dire à faire voyager des informations dans l'espace et dans le temps.

Toutefois, cette évolution ne signe pas pour autant la fin de la notion d'algorithme, car, pour transmettre de l'information sur les réseaux ou retrouver

de l'information dans une base de données, il faut inventer de nouveaux algorithmes. Ainsi, les algorithmes de routage, qui permettent d'orienter les informations qui voyagent sur les réseaux, et les algorithmes de notation de pages, à l'œuvre dans les moteurs de recherche, sont aujourd'hui parmi les algorithmes les plus étudiés. De même, pour interroger les bases de données, nous avons développé de nouveaux langages : les langages de requêtes. Ce sont donc encore les quatre mêmes concepts qui sont au centre de ces nouvelles questions de transmission et de stockage des informations.

On peut regretter qu'aucune langue n'ait réussi à exprimer ce caractère synthétique de l'informatique. Certaines langues, comme l'anglais, en font la science des machines, d'autres langues, comme le français, la science de l'information. Les unes comme les autres désignent le tout par la partie.

L'informatique et les autres sciences

L'apparition de l'informatique a quelque peu bouleversé notre vision de l'organisation des sciences. La classification des sciences qui a prévalu jusqu'au début du XX^e^ siècle opposait les mathématiques aux sciences de la nature. Sur le plan ontologique, les mathématiques cherchent à découvrir des vérités nécessaires, alors que les sciences de la nature cherchent à découvrir des vérités contingentes. Sur le plan méthodologique, les mathématiques, qui ne demandent aucune interaction avec la nature, sont une science *a priori*, alors que les sciences de la nature, fondées sur l'observation et l'expérimentation, sont des sciences *a posteriori*. L'informatique n'appartient à aucune de ces deux catégories, car elle est du côté des mathématiques sur le plan ontologique, mais du côté des sciences de la nature sur un plan méthodologique : le fait qu'un algorithme de tri transforme la suite 2, 8, 1, 5 en la suite 1, 2, 5, 8 est une vérité nécessaire, qui ne doit rien aux propriétés de la nature ; en revanche, pour l'établir, nous interagissons avec un ordinateur, qui est un objet de la nature.

Mais c'est surtout parce que les quatre concepts d'algorithme, de machine, de langage et d'information se sont propagés dans l'ensemble des sciences, que l'informatique a transformé le paysage scientifique. La notion d'information, par exemple, a permis aux physiciens de rétrospectivement mieux comprendre la notion d'entropie, qui se définit désormais comme l'information sur un système qui manque dans sa description macroscopique. Comme nous le verrons, c'est cette même notion d'information qui permet de définir la notion d'aléa.

Les notions d'algorithme et de langage, par ailleurs, se sont révélées de formidables outils de description d'objets concrets et abstraits, en particulier dans des situations, comme celle de la description des systèmes biologiques, où

le traditionnel langage des équations différentielles se révélait inopérant. On peut, par exemple, décrire la manière dont une cellule fabrique une protéine à partir d'un brin d'ARN par un algorithme qui associe un texte écrit dans un alphabet de vingt lettres – la protéine – à tout texte écrit dans un alphabet de quatre lettres – le brin d'ARN. De même, la logique définit aujourd'hui les démonstrations comme des algorithmes, par exemple une démonstration de la proposition $A \Rightarrow B$ comme un algorithme qui transforme les démonstrations de A en des démonstrations de B. Et si les scientifiques et les ingénieurs, jusqu'au début du XX^e^ siècle, ont décrit les grammaires ou les réseaux de chemin de fer en langue naturelle, ils inventent désormais chaque jour de nouveaux langages formels pour le faire de manière plus précise.

Enfin, s'il est devenu banal de s'émerveiller de la manière dont l'informatique a transformé l'instrumentation scientifique, il est peut-être moins fréquent de remarquer que les mathématiques étaient, jusqu'aux années 70, la seule science à ne pas utiliser d'instruments – à quelques rares exceptions près, comme la règle et le compas – jusqu'à ce que les ordinateurs les fassent entrer dans la phase instrumentée de leur histoire.

La complexité

Quand on démonte un réveil, une locomotive à vapeur ou un poste de radio, on se retrouve, en général, avec des engrenages, des tubes à fumée ou des composants électroniques, qui se comptent en dizaines ou en centaines.

Un programme informatique, en revanche, est souvent composé de plusieurs centaines de milliers d'instructions, voire de plusieurs millions pour certains. Le développement de programmes informatiques a donc constitué un saut radical dans la complexité des objets que nous sommes capables d'étudier ou de construire. Ces programmes ont permis en retour un saut similaire dans la complexité d'autres objets : grâce aux ordinateurs, nous sommes aujourd'hui capables de construire des circuits électroniques formés de plusieurs centaines de millions de transistors, des démonstrations mathématiques formées de plusieurs millions de pages ou des catalogues de plusieurs millions de molécules.

Une science et une technique

L'apparition de l'informatique nous mène aussi à revoir les relations qui existent entre les sciences et les techniques. Nous étions accoutumés à voir les techniques comme des applications des sciences, ce que traduit notre habitude de les appeler « sciences appliquées ». Cette notion d'application suppose que les sciences déterminent les techniques, lesquelles, en revanche, ne les influencent nullement. Depuis le XVII^e^ siècle, cette vision des relations entre

sciences et techniques est en contradiction avec les faits, ne serait-ce qu'à cause de l'importance des instruments dans la démarche expérimentale. Mais nous évacuons souvent cette objection en minimisant l'importance des instruments, que nous qualifions d'« outils », ce qui nous permet de rester cohérents avec nos valeurs qui, depuis l'Antiquité, placent le savoir au-dessus du faire.

Il est probable que l'apparition de l'informatique nous contraigne à abandonner cette hiérarchie. Ignorer que l'informatique étudie des objets – machines, programmes, etc. – mais en fabrique également donne une image complètement déformée de la discipline. Il est certes possible de distinguer une activité scientifique, qui vise à connaître, d'une activité technique, qui vise à fabriquer, mais il n'est plus possible pour le savant d'ignorer l'activité de l'ingénieur.

Comprendre cela est essentiel pour qui enseigne l'informatique. Enseigner l'informatique demande de faire sienne cette phrase attribuée à Confucius : « J'entends et j'oublie, je vois et je me souviens, je fais et je comprends. » Enseigner l'algorithmique sans enseigner aussi à écrire un programme ou comment on construit un ordinateur avec des transistors, c'est transmettre un savoir amputé.

Enseigner l'informatique

Comme celui de toutes les disciplines, l'enseignement de l'informatique a profondément évolué au cours de ces vingt dernières années, du fait de l'apparition de gigantesques ressources en ligne. Au XX^e^ siècle, on enseignait la représentation numérique des caractères en expliquant que le code ASCII de la lettre « A » était 65, que celui de la lettre « B » était 66, etc. On enseignait la programmation en expliquant que l'on testait la parité d'un nombre entier avec la fonction `odd` et que l'on calculait l'arc tangente d'un nombre réel avec la fonction `atn`.

Aujourd'hui, toutes ces informations factuelles se retrouvent sans peine sur le Web. Enseigner ne consiste plus à transmettre des atomes de connaissances, mais à les mettre en perspective et à montrer comment ils s'assemblent.

Un parcours

Notre parcours commence avec la notion d'information : nous montrons comment les objets les plus divers se numérisent et comment l'information se quantifie. Nous poursuivons au deuxième chapitre avec la notion de langage de programmation en insistant sur la difficulté, mais l'importance, de trouver des mots pour expliquer avec précision ce qu'il se passe quand on exécute un programme. Dans le troisième chapitre, nous présentons un certain nombre

d'algorithmes classiques qui peuvent s'exprimer dans un tel langage et nous insistons sur la multiplicité des algorithmes qui permettent de résoudre un problème donné. Au quatrième chapitre, nous expliquons enfin la manière dont un ordinateur se construit étape par étape à partir du transistor et la manière dont un programme écrit dans un langage évolué se traduit en langage machine, le langage natif de l'ordinateur, déterminé par son architecture. Nous poursuivons au cinquième chapitre avec la notion de réseau, en particulier le réseau Internet, et la manière dont des algorithmes permettent de transmettre de l'information d'un bout à l'autre de ces machines. Enfin, dans les sixième et septième chapitres, nous revenons sur la notion d'information, abordée du point de vue de sa protection et de sa structuration, puis des bases de données. Parce que le Web est une base de données et parce qu'il est l'interface de nombreuses autres bases de données, c'est aussi dans ce chapitre que le Web est abordé.

Bien entendu, de nombreux sujets ne sont pas abordés dans ce bref parcours. Si celui-ci donne envie au lecteur de poursuivre le voyage et de lire quelques-unes des références que nous donnons à la fin de chaque chapitre, nous aurons atteint notre but.