

Algorithmique

1. Introduction

1

Plan du cours

- I. Introduction
- II. Variables
- III. Structures alternatives
- IV. Structures itératives
- V. Tableaux
- VI. Fonctions et procédures
- VII. Fichiers

▶ 2

Algorithmique ESI 2023-2024

2

Plan du cours

- I. Introduction**
- II. Variables
- III. Structures alternatives
- IV. Structures itératives
- V. Tableaux
- VI. Fonctions et procédures
- VII. Fichiers



3

Algorithmique

ESI

2023-2024

3

Sommaire

- I. Définitions
- II. Historique
- III. Concepts de base
- IV. Structure générale d'un algorithme



4

Algorithmique

ESI

2023-2024

4

Définitions

5

Définitions

- ▶ Algorithme : **suite d'instructions** qui, si elles sont exécutées correctement, permettent de résoudre un problème donné
- ▶ Ces instructions doivent être **compréhensibles** par celui qui va exécuter l'algorithme

6

Définitions

▶ Algorithme

- ▶ N'est pas réservé à l'informatique
- ▶ Est souvent utilisé dans la vie quotidienne
 - ▶ Manuels d'utilisation des équipements électroniques ;
Recettes de cuisine ; Instructions de GPS ; Notices des médicaments ...

▶ 7

Algorithmique

ESI

2023-2024

7

Définitions

▶ Algorithme

- ▶ En informatique, un algorithme est une **suite ordonnée** d'**instructions** qui **décrit de façon exhaustive** les **différentes étapes** à suivre par un **processeur** pour **résoudre un problème** donné en un **temps fini**

▶ 8

Algorithmique

ESI

2023-2024

8

Historique

9

Historique

- ▶ Les premiers algorithmes connus étaient utilisés à Babylone dès l'an 2000 BC environ pour effectuer des calculs avancés des impôts
- ▶ Euclide a conçu en l'an 300 BC environ un algorithme pour calculer le plus grand commun diviseur (PGCD) de deux nombres
- ▶ Archimède a proposé en l'an 250 BC environ un algorithme pour le calcul de Π

10

Historique

- ▶ Le mathématicien musulman Muhammad Al-Khwarizmi, au début du 8^e siècle, est le premier à avoir systématiser des algorithmes en proposant plusieurs algorithmes pour la résolution d'équations du second degré
- ▶ Le mot **Algorithme** trouve son origine dans le nom latinisé " *Algoritmi* " d'Al-Khwarizmi

▶ I1

Algorithmique

ESI

2023-2024

11

Historique

- ▶ Averroès, au 12^e siècle, propose une méthode de raisonnement itérative, où la thèse s'affine d'étape en étape
- ▶ Adelard de Bath, mathématicien britannique, a introduit au 12^e siècle le terme latin *algorismus* par référence au nom d'Al-Khawarizmi. Ce mot donnera lieu plus tard au mot *algorithme*

▶ I2

Algorithmique

ESI

2023-2024

12

Historique

- ▶ Ada Lovelace, mathématicienne britannique, a réalisé en 1843 la première implémentation d'un algorithme (pour le calcul des nombres de Bernoulli) sous forme de programme

▶ I3

Algorithmique

ESI

2023-2024

13

Historique

- ▶ Alan Turing, mathématicien britannique, formalise les concepts d'algorithme et de calculabilité en inventant la *machine de Turing* en 1936. Cette machine virtuelle donne une définition précise au concept d'algorithme et permet de résoudre tout problème de calcul basé sur une procédure algorithmique

▶ I4

Algorithmique

ESI

2023-2024

14

Concepts de base

15

Concepts de base

- ▶ L'ordinateur est une machine qui est
 - ▶ Incapable d'initiative → ne peut exécuter une tâche que si on lui fournit le détail des instructions à réaliser, i.e. que si on le programme
 - ▶ Capable de stocker un **programme** et de l'exécuter

16

Concepts de base

- ▶ Un **programme** informatique est l'implémentation d'un ou plusieurs **algorithme(s)** dans un **langage de programmation** donné
- ▶ Ainsi, tout programme a, par définition, la structure d'un algorithme
- ▶ La **programmation** est la tâche d'écrire un programme

▶ I7

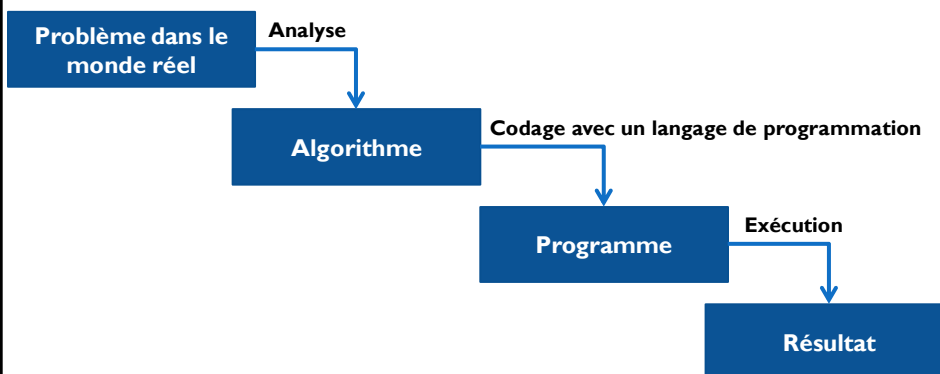
Algorithmique

ESI

2023-2024

17

Concepts de base



▶ I8

Algorithmique

ESI

2023-2024

18

Concepts de base

- ▶ Un **langage de programmation** est une notation conventionnelle destinée à implémenter des algorithmes afin de produire des programmes informatiques
- ▶ À l'instar des langues naturelles, il est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations

▶ 19

Algorithmique ESI 2023-2024

19

Concepts de base

- ▶ Niveaux des **langages de programmation**
 - ▶ Langages de **bas niveau** : sont très loin de la langue naturelle (i.e. difficile à comprendre par les humains) et fournissent très peu d'abstraction par rapport aux instructions du processeur de la machine

▶ 20

Algorithmique ESI 2023-2024

20

Concepts de base

► Niveaux des **langages de programmation**

- Langages de **haut niveau** : sont proches de la langue naturelle (i.e. facile à comprendre par les humains) et permettent un grand niveau d'abstraction avec l'utilisation de mots usuels (*print, read, save, when, if...else, ...*) et de symboles de mathématiques

► 21

Algorithmique

ESI

2023-2024

21

Concepts de base

► Langages de programmation de **bas niveau**

- **Langage machine** : appelé aussi **code machine**, c'est un ensemble d'instructions et de données codées en binaire (des bits de 0 ou 1) interprété par le processeur exécutant le programme
 - Il s'agit du langage natif d'un processeur, i.e. le seul qu'il puisse comprendre et traiter
 - Il est propre à chaque processeur

► 22

Algorithmique

ESI

2023-2024

22

Concepts de base

```

00000000 fc 31 c0 8a c0 8a d8 8e d0 bc 00 7c 89 e6 bf 00 |.l.....|...|
00000010 06 b9 00 01 f3 a5 89 fd b1 08 f3 ab fe 45 f2 e9 |...F...x...N.@|
00000020 00 8a f6 46 bb 20 75 08 84 d2 78 07 80 4e bb 40 |...V...R...l...|
00000030 8a 56 ba 88 56 00 e8 fc 00 52 bb c2 07 31 d2 88 |o...V.s.....|
00000040 6f fc 0f a3 56 bb 73 19 8a 07 bf 87 07 b1 03 f2 |.t.....F.....|
00000050 ae 74 0e b1 0b f2 ae 83 c7 09 8a 0d 01 cf e8 c5 |.B...s.X...u.r.|
00000060 02 40 82 c3 10 73 d8 58 2c 7f 3a 06 75 04 72 05 |Ht.O...F.....|
00000070 48 74 0d 30 c0 04 b0 88 46 b8 bf b2 07 e8 a6 00 |.f.....V.N.....|
00000080 be 7b 07 e8 b2 00 8a 56 b9 4e e8 8e 00 eb 05 b0 |...0.....|
00000090 07 e8 b0 00 30 e4 cd 1a 89 d7 03 7e bc b4 01 cd |.u.O...9.r...F...|
000000a0 16 75 0d 30 e4 cd 1a 39 fa 72 f2 8a 46 b9 eb 16 |o...<.t...<.v.|
000000b0 30 e4 cd 16 88 e0 3c 1c 74 f1 2c 3b 3c 04 76 06 |<.w...F.s...F.|
000000c0 2c c7 3c 04 77 c9 98 0f a3 46 0c 73 c2 88 46 b9 |...<.t.....|
000000d0 be 00 08 8a 14 89 f3 3c 04 9c 74 0a c0 e0 04 05 |...S.F.@...|
000000e0 be 07 93 c6 07 80 53 f6 46 bb 40 75 08 bb 00 06 |...Y.^..u...V...0..|
000000f0 b4 03 e8 59 00 5e 9d 75 06 8a 56 b8 80 ea 30 bb |...G.r...U...|
00000100 00 7c b4 02 e8 47 00 72 86 81 bf fe 01 55 aa 0f |...F.$...|
00000110 85 7c ff be 85 07 e8 19 00 ff e3 b0 46 e8 24 00 |.l.....V..x...|
00000120 b0 31 00 d0 eb 17 0f ab 56 0c be 78 07 e8 ab ff |...u...|
00000130 89 fe e8 03 00 be 85 07 ac a8 80 75 05 e8 04 00 |...$ S.....[.t|
00000140 eb f6 24 7f 53 bb 07 00 b4 0e cd 10 5b c3 8a 74 |.L...V...F..t.f|
00000150 01 8b 4c 02 b0 01 56 89 e7 f6 46 bb 80 74 13 66 |j.f.t..Sj.j...H.|
00000160 6a 00 66 ff 74 08 0e 53 6a 01 6a 10 89 e6 48 80 |.e...^...Defa|
00000170 cc 40 cd 13 89 fe 5e c3 20 20 a0 0a 44 65 66 61 |ult:...|
00000180 75 6c 74 3a a0 0d 8a 00 05 0f 01 06 07 0b 0c 0e |...|
00000190 83 a5 a6 a9 0d 0c 0b 0a 09 08 0a 0e 11 10 01 3f |...?|
000001a0 bf 44 4f d3 4c 69 6e 75 f8 46 72 65 65 42 53 c4 |.D.O.Linu.FreeBS.|
000001b0 66 bb 44 72 69 76 65 20 00 00 80 8f b6 00 00 00 |f.Drive.....|
000001c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 55 aa |.....U..|
00000200

```

► Exemple d'un
fichier binaire
exécutable
(hexadécimal)

► 23

Algorithmique

ESI

2023-2024

Par Original téléversé par Scientus sur Wikipédia anglais. — see above, BSD, <https://commons.wikimedia.org/w/index.php?curid=7965031>

23

Concepts de base

► Langages de programmation de **bas niveau**

► **Langage assembleur** : appelé aussi **langage d'assemblage**, il représente le langage machine sous une forme lisible par l'humain

► Il est propre à chaque processeur

► 24

Algorithmique

ESI

2023-2024

24

Concepts

```

MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2

C000                                ORG    ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START              LDS    #STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROY: acc A

0013      RESETA EQU    400010011
0011      CTLREG EQU    400010001

C003 86 13      INITA  LDA A  #RESETA  RESET ACIA
C005 87 80 04          STA A  ACIA
C008 86 11          LDA A  #CTLREG  SET 8 BITS AND 2 STOP
C00A 87 80 04          STA A  ACIA
C00D 7E C0 F1          JMP    SIGNON  GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROY: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

C010 86 80 04  INCH  LDA A  ACIA      GET STATUS
C013 47          AER A                SHIFT RCRP FLAG INTO CARRY
C014 24 FA      BCC  INCH  RECEIVE NOT READY
C016 86 80 05  LDA A  ACIA+1  GET CHAR
C019 84 7F      AND A  #$7F  MAKE PARITY
C01B 7E C0 79  JMP  OUTCH  ECHO & RTS

*****
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: digit in acc A
* CALLS: INCH
* DESTROY: acc A
* Returns to monitor if not HEX input

C01E 8D F0  INHEX  BSR  INCH  GET A CHAR
C020 81 30      CMP A  #'0  ZERO
C022 2B 11      BMI  HEXERR  NOT HEX
C024 81 39      CMP A  #'9  NINE
C026 2F 0A      BLS  HEXRST  GOOD HEX
C028 81 41      CMP A  #'A
C02A 2B 09      BMI  HEXERR  NOT HEX
C02C 81 46      CMP A  #'F
C02E 2E 05      BGT  HEXERR
C030 80 07      SUB A  #7    FIX A-F
C032 84 0F      HEXRST AND A  #$0F  CONVERT ASCII TO DIGIT
C034 39          RTS

C035 7E C0 AF  HEXERR JMP  CTRL  RETURN TO CONTROL LOOP

```

▶ 25
Algorithmique
ESI
2023-2024

Par Swtpc6800 en:User:Swtpc6800 Michael Holley — Travail personnel, Domaine public, <https://commons.wikimedia.org/w/index.php?curid=15345825>

25

Concepts de base

- ▶ Langages de programmation de **haut niveau**
 - ▶ Il existe des centaines de langages de programmation de haut niveau

▶ 26
Algorithmique
ESI
2023-2024

26

Concepts de base

Worldwide, Sept 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	27.99 %	+0.1 %
2		Java	15.9 %	-1.1 %
3		JavaScript	9.36 %	-0.1 %
4		C#	6.67 %	-0.4 %
5		C/C++	6.54 %	+0.3 %
6		PHP	4.91 %	-0.4 %
7		R	4.4 %	+0.2 %
8		TypeScript	3.04 %	+0.2 %
9	↑↑	Swift	2.64 %	+0.6 %
10		Objective-C	2.15 %	+0.1 %

► 27

Popularity of Programming Languages <https://pypl.github.io/PYPL.html> Algorithmique ESI 2023-2024

27

Concepts de base

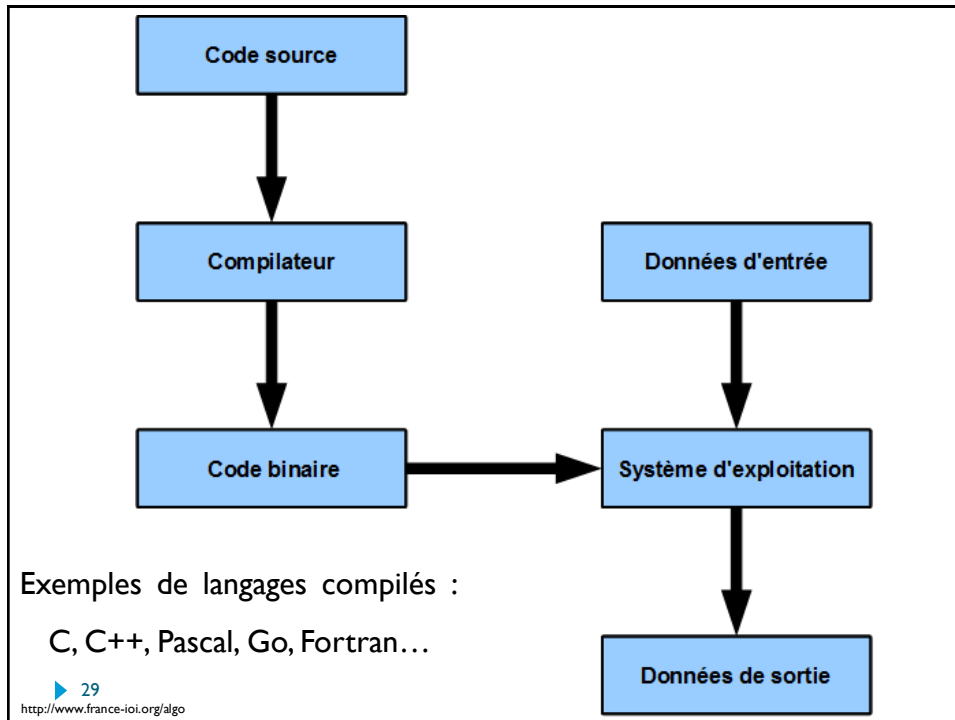
► Langages de programmation de **haut niveau**

- Nécessitent une traduction vers le langage machine par un **compilateur** (programme traduisant l'ensemble des instructions demandées dans un langage de bas niveau, notamment assembleur ou machine) ou un **interpréteur** (programme exécutant pas à pas les instructions demandées)

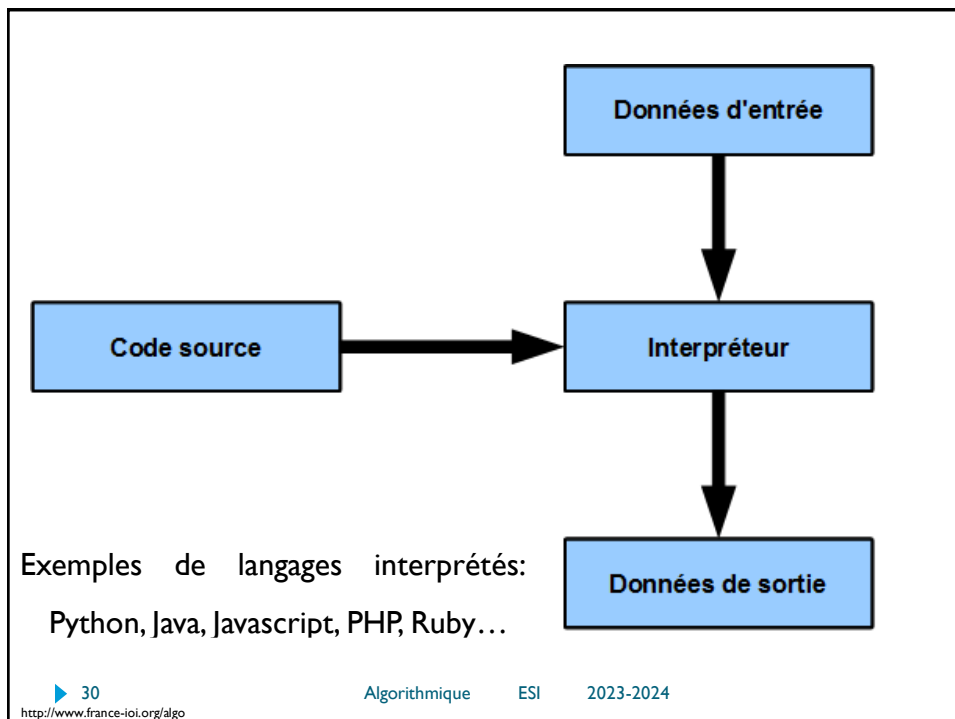
► 28

Algorithmique ESI 2023-2024

28



29



30

Concepts de base

- ▶ Langages de programmation de **haut niveau**
 - ▶ Un langage interprété permet la portabilité, i.e. obtenir un code source qui fonctionne sur tout ordinateur
 - ▶ Un langage compilé permet d'exécuter le programme directement sur l'ordinateur → plus rapide que le même programme dans un langage interprété

Structure générale d'un algorithme

Écriture d'un algorithme

- ▶ Un algorithme s'adresse à un être humain qui va généralement le traduire dans un langage de programmation
- ▶ Son écriture doit détailler le plus possible son fonctionnement et sa structure
 - ▶ Entrées; Enchaînement des instructions; Structures de contrôle; Sorties

▶ 33

Algorithmique

ESI

2023-2024

33

Écriture d'un algorithme

- ▶ Un algorithme est **indépendant** du langage de programmation
- ▶ Il y a plusieurs possibilités pour écrire un algorithme
 - ▶ Utilisation de phrases dans la langue naturelle
 - ▶ Représentation graphique à l'aide d'organigrammes, logigrammes ou algorigrammes
 - ▶ Utilisation de **pseudo-code**

▶ 34

Algorithmique

ESI

2023-2024

34

Écriture d'un algorithme

- ▶ Très souvent, un algorithme est écrit en **pseudo-code**
 - ▶ Forme proche de la langue humaine, avec des conventions proches des langages de programmation mais sans les contraintes syntaxiques de ces derniers
 - ▶ Utilisation de mots-clés

ALGORITHME	PROCEDURE	FONCTION	VAR	CONSTANTE
PARAM	DEBUT	FIN	SI	SINON
SELON	POUR	AUTREMENT	TANT_QUE	REPETER
JUSQU'À	OU	ET	NON	...

35

Écriture d'un algorithme

- ▶ Avantages de l'utilisation du pseudo-code
 - ▶ Lisibilité
 - ▶ Surtout avec l'utilisation des commentaires
 - ▶ Facilitation du passage au langage de programmation
 - ▶ Unification de l'écriture des algorithmes

36

Écriture d'un algorithme

- ▶ Structure générale d'un algorithme écrit en pseudo-code
 - ▶ **En-tête** : Commence par le mot-clé ALGORITHME et permet de définir le nom de l'algorithme
 - ▶ **Partie déclarative** : Permet de déclarer les différents objets utilisés dans l'algorithme
 - ▶ **Corps** : Est délimitée par les mots-clés DEBUT et FIN et contient toutes les instructions de l'algorithme

▶ 37

Algorithmique

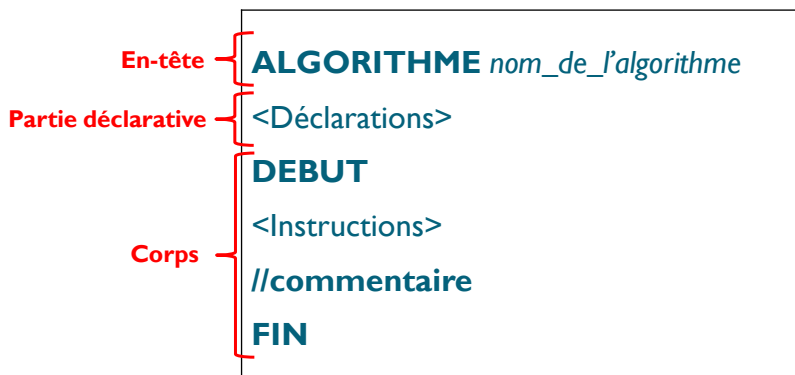
ESI

2023-2024

37

Écriture d'un algorithme

- ▶ Structure générale d'un algorithme écrit en pseudo-code



▶ 38

Algorithmique

ESI

2023-2024

38

Écriture d'un algorithme

► Exemple

ALGORITHME calcul_du_périmètre

VAR longueur, largeur, p : réel

DEBUT

Afficher("Donner la longueur et la largeur: ")

Lire(longueur, largeur)

$p \leftarrow 2 * (\text{longueur} + \text{largeur})$

Afficher("Le périmètre est : ", p)

FIN

► 39

Algorithmique

ESI

2023-2024

39

Des idées claires et distinctes

Luc de Brabandere

Dans cette chronique mensuelle, Luc de Brabandere, philosophe d'entreprise et fondateur de l'agence Cartoindus, analyse un mot utilisé couramment dans les entreprises.



Algorithme

Je sais, le mot est un peu rébarbatif. Ce pourrait être le nom d'une figure musicale particulièrement archaïque réservée aux meilleurs pianistes, ou celui d'un médicament à base de sel marin destiné à stabiliser les pulsations cardiaques. Mais ce n'est pas le cas.

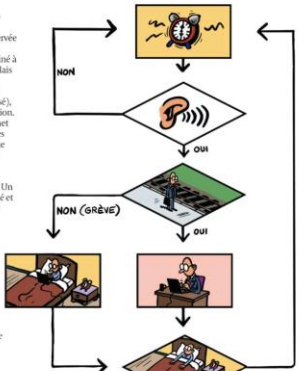
Fidèle à l'esprit de cette chronique (dont l'algorithmique est le cinquième mot analysé), commençons donc par proposer une définition. Un algorithme est une méthode qui permet d'atteindre un but en un nombre fini d'étapes distinctes. Concrètement ou non, c'est ce que nous faisons toute la journée. Pour tondre le gazon, trouver du stop-défilé dans un supermarché ou choisir une destination de vacances, nous avons notre manière de faire. Un algorithme est un mode d'emploi formelisé et disponible qui évite de devoir repartir à zéro chaque fois qu'un problème ou un souhait apparaît. Il peut être personnel, culturel ou universel.

Qui dit méthodes dit ordos, c'est-à-dire chemin. Ce n'est pas un hasard, car utiliser un algorithme, c'est suivre un parcours fléché et progresser par à-coups. C'est une forme de savoir codé, structuré, mémorisé, qui peut donc être réutilisé ou transmis. La recette du tiramisu est un algorithme, tout comme l'est la séquence d'instructions qui permet de décaler une machine à café.

Si, alors...

Ces deux exemples du monde de la cuisine illustrent le cas le plus simple, celui de l'algorithme en ligne droite dont on parcourt consciencieusement toutes les étapes, les unes après les autres. Le chemin est le même pour tout le monde, et on atteint son objectif sans trop se poser de questions. Mais cette situation est finalement rare, car le plus souvent, dès qu'on veut atteindre un but, précisément, de nombreuses questions se posent.

Pour définir la meilleure manière d'aller de



Louxor au sommet du Mont Blanc, tout dépendra du temps et du budget dont vous disposez, de vos convictions écologiques, des conditions sanitaires et politiques, de votre tempérament et de votre condition physique, des personnes qui voyagent éventuellement avec vous, etc. La question de départ va en générant des dizaines d'autres, qui bien souvent seront de type «si, alors».

La plupart des algorithmes sont truffés de ce type de fourchettes qui impliquent autant de choix à faire. On le voit en remplissant sa feuille d'impôts : «si vous n'avez pas de revenus immobiliers, alors passer à la case 7», etc.

Mais la logique des «si» et des «ou» réserve des surprises. Pour vous en convaincre, il vous suffit de comparer les deux raisonnements suivants...

Si j'ai 18 ans, je peux voter.
Si j'ai 18 ans, je peux voter et conduire.

Si j'ai 4 euros, je peux acheter un journal.
Si j'ai 4 euros, je peux acheter un sandwich.

Donc, si j'ai 4 euros, je peux acheter un journal et un café.

Convergence
Il en va des algorithmes comme de n'importe quelle méthode, certains sont plus efficaces que d'autres.

Même dans le cas le plus simple, quand aucune question de type «si, alors» ne se pose, certains algorithmes vont plus vite que d'autres. Les mathématiciens disent qu'ils «convergent» plus rapidement vers l'objectif souhaité.

En 1674, Leibniz proposa une formule extrêmement simple pour calculer pi.

$$\pi = 4 \left(1 - \frac{1}{5} + \frac{1}{9} - \frac{1}{13} + \dots \right)$$

La méthode est correcte, mais la convergence est désespérément lente, 300 termes sont nécessaires pour obtenir seulement... deux décimales, le fameux 3,14 ! À une époque où la machine à calculer n'existait pratiquement pas, on comprend que la méthode n'eût pas beaucoup de succès. La science des algorithmes est entre autres celle de leur efficacité. Et les progrès faits dans le calcul du nombre pi sont un bel exemple de la créativité des mathématiciens dans la conception des algorithmes.

Dans les cas plus compliqués des algorithmes à bifurcations, la question de la convergence se pose de manière plus agitée encore.

Pour trier alphabétiquement cinq cents cartes de visite, une méthode pourrait consister à les prendre une à une, et à les classer chaque fois au bon endroit, en partant de la première jusqu'à la dernière. Mais cela ira plus vite en commençant par un premier tri basé sur la première lettre, et trier ensuite chacun des paquets séparément, avant de les mettre, un paquet à la fois, dans l'ordre alphabétique.

Explosion combinatoire

Dans les algorithmes à fourche «si-alors», le nombre de possibilités devient très vite immense, on parle alors d'explosion combinatoire. Si au Scrabble, un algorithme peut encore affirmer avec certitude quel est le meilleur coup à jouer, aux échecs par contre ce n'est plus possible. Si j'avance ce pion, alors elle va bouger sa tour, alors je vais déplacer ma reine... Mais si j'avance pas mon pion, alors elle... Le «si, alors», devient vite «si, alors, alors, alors...», et l'algorithme deviendrait fou.

Ceci nous amène à une troisième distinction, cette fois entre les algorithmes pour lesquels un objectif précis peut être atteint, et les autres qui ne peuvent jamais que s'en rapprocher. Le bot n'est plus alors d'optimiser, mais bien de trouver la solution la plus satisfaisante possible. Le pèlerin de Compostelle qui choisit ce qu'il met dans son sac à dos ou la librairie qui veut rendre son étalage le plus attrayant possible sont dans cette situation. Il ne leur sera jamais possible de savoir avec certitude s'ils ont fait le meilleur choix.

Vous l'aurez remarqué, j'arrive au bout de cette chronique consacrée aux algorithmes, sans avoir pour autant utilisé les mots «ordinateurs» ou «informatique». Ce n'est évidemment pas un hasard.

40

Algorithmique

1. Introduction