

Introduction à la Programmation des Algorithmes

1.1. Introduction – Informatique et ordinateurs

François Fleuret

<https://fleuret.org/11x001/>



**UNIVERSITÉ
DE GENÈVE**

Objectifs de ce cours

1. Connaître les bases des langages C et Python.
2. Savoir analyser et décomposer un problème pour concevoir un **algorithme**.
3. Connaître les principaux outils qui existent pour faciliter la programmation.

L'informatique

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.
- Synthèse d’images, jeux vidéos.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.
- Synthèse d’images, jeux vidéos.
- Simulations physiques, prédictions météorologiques.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.
- Synthèse d’images, jeux vidéos.
- Simulations physiques, prédictions météorologiques.
- Cryptographie.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.
- Synthèse d’images, jeux vidéos.
- Simulations physiques, prédictions météorologiques.
- Cryptographie.
- Guidage, assistance au pilotage de véhicules.

Informatique

“Science du traitement automatique et rationnel de l’information considérée comme le support des connaissances et des communications.” (Larousse)

Les tâches que traitent les techniques informatiques sont extrêmement diverses:

- Comptabilité, bases de données, création de documents.
- Création et transmission de médias.
- Contrôle et optimisation de processus industriels.
- Synthèse d’images, jeux vidéos.
- Simulations physiques, prédictions météorologiques.
- Cryptographie.
- Guidage, assistance au pilotage de véhicules.
- Intelligence artificielle: reconnaissance de la parole ou d’images, optimisation de stratégies.

L'informatique n'est pas liée à un type d'ordinateur ou une technologie particulière.

L'informatique n'est pas liée à un type d'ordinateur ou une technologie particulière.

"Computer science is no more about computers than astronomy is about telescopes." – Edsger W. Dijkstra

"I don't need to waste my time with a computer just because I am a computer scientist." – Edsger W. Dijkstra

- Comprendre les principes des langages de programmation et savoir utiliser ces langages.
- Analyser et décomposer un problème.
- Créer l'**algorithme** adéquate,
 - estimer sa “classe de complexité”;
 - expliquer son comportement et prévoir ses résultats;
 - l'optimiser.

Ce cours traite à la fois de:

- **Connaissances déclaratives:** “de quoi s’agit-il? quel est le problème?”
- **Connaissances opératoires:** “comment résoudre opérationnellement le problème posé?”
- Leur mise en œuvre à un niveau d’abstraction élevé.

Ordinateurs

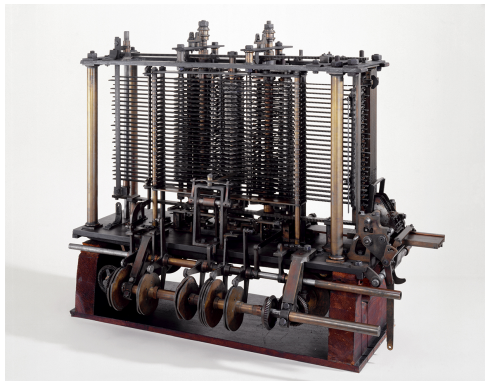


Métier Jacquard



Joseph Marie Jacquard
(1752–1834)

Des cartes perforées en carton encodent des motifs et un mécanisme pilote le métier à tisser.

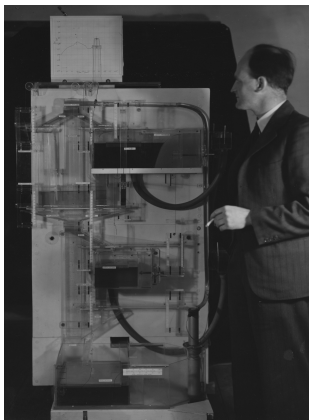


Machine de Babbage (1834)



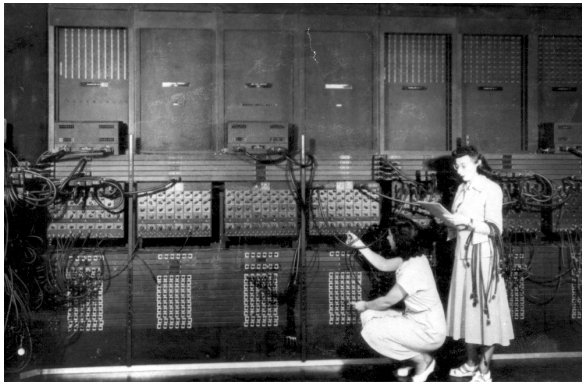
Lady Ada Lovelace
(1815–1852)

La machine de Babbage possède des registres de mémoire, et les traite selon des instructions encodées sur des cartes perforées. Ada Lovelace dans une note de 1843 décrit un programme pour cette machine pour calculer les nombres de Bernoulli.



MONIAC (Monetary National Income Analogue Computer, 1949)

Cette machine utilisait un circuit hydraulique pour faire des calculs de nature économique. Les niveaux dans les différents réservoirs correspondaient à des budgets, et les échanges pouvaient être modulés pour simuler des politiques budgétaires.



ENIAC (Electronic Numerical Integrator and Computer, 1945)

Premier ordinateur électronique et programmable. Poids total 30t, \simeq 400 opérations par seconde, mémoire de 20 nombres et cartes perforées.

172

9/9

0800 Antan started
 1000 stopped - antan ✓

		{ 1.2700	9.032 847 025
13 ⁰⁰ (033)	MP-MC	2.130476415	9.037 846 895 correct
(033)	PRO	2.130476415	4.615925059(-2)
	conv	2.130676415	

Relays 6-2 in 033 failed special speed test
 in relay 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)
 1525 Started Multi Adder Test.

1545 Relay #70 Panel F
 (moth) in relay.

1630/1630 First actual case of bug being found.
 antan started.
 1700 closed down.

Relay 2145
 Relay 3376



Grace Hopper
 (1906–1992)

Premier “bug”

Grace Hopper dirigeait une équipe d'ingénieurs qui développa le premier compilateur en 1954, vu comme un précurseur du COBOL, qui est toujours utilisé.

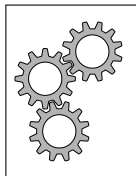
“A computer is like a mischievous genie. It will give you exactly what you ask for, but not always what you want.” – Joe Sondow

Fondamentalement, un ordinateur manipule des séries de chiffres qui sont interprétées de manières différentes en fonction de l'opération et du contexte:

- Instructions élémentaires à exécuter.
- Caractères, textes.
- Nombres entiers ou à virgules, vecteurs, matrices, courbes.
- Images, sons.

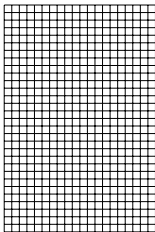
Structure d'un ordinateur

Microprocesseur (CPU)

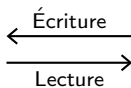
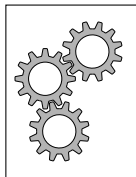


Structure d'un ordinateur

Mémoire vive (RAM)

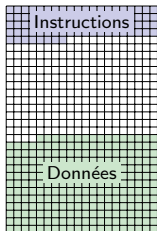


Microprocesseur (CPU)

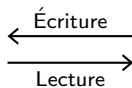
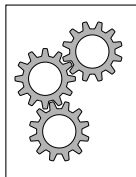


Structure d'un ordinateur

Mémoire vive (RAM)

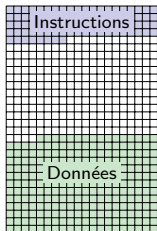


Microprocesseur (CPU)

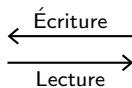
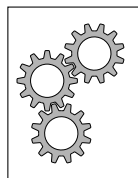


Structure d'un ordinateur

Mémoire vive (RAM)

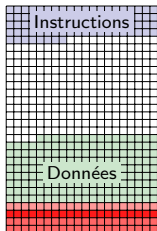


Microprocesseur (CPU)

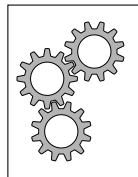


Structure d'un ordinateur

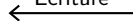
Mémoire vive (RAM)



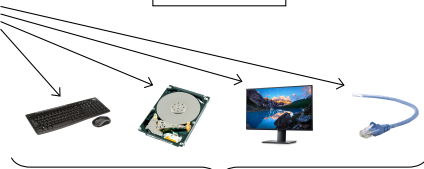
Microprocesseur (CPU)



Écriture



Lecture



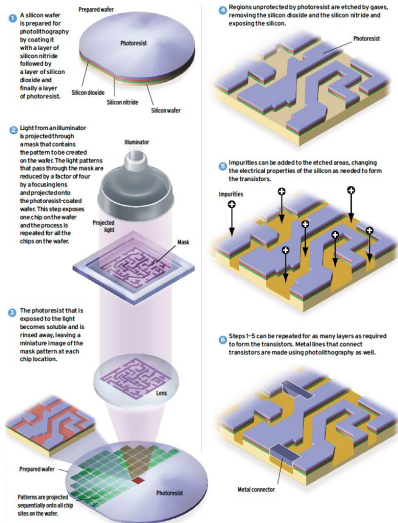
Périphériques

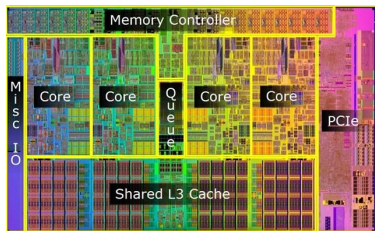
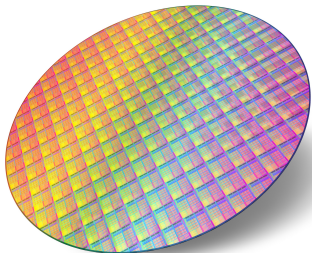
L'interaction avec la **mémoire vive** se fait dans un programme comme s'il s'agissait d'une longue liste de valeurs, chacune avec une adresse.

Plusieurs valeurs peuvent être groupées pour représenter une quantité ou un objet complexe, nous y reviendrons.

L'interaction avec les **mémoires de masses** telles que les disques durs se fait en accédant à des **fichiers**, chacun ayant une structure similaire à la mémoire vive (séquence de valeurs).

La technologie standard pour faire des microprocesseurs est la VLSI (Very large-scale integration) qui utilise la **photolithographie**.





Ordres de grandeur

- **Vitesse de calcul:** $\simeq 10^{10}$ (CPU), $\simeq 10^{14}$ (GPU) opérations arithmétiques par seconde.
- **Mémoire vive:** $\simeq 10^{10}$ caractères ($\simeq 10^7$ pages de texte, 7 jours de musique, 9h de vidéos).
- **Mémoire de masse:** $\simeq 10^{12}$ caractères ($\simeq 10^9$ pages de texte, 2 années de musique, 1 mois de vidéos).
- **Vitesse de communication:** $\simeq 10^8$ caractères par seconde (fibre optique).

Exemple de calcul intensif: Produit de deux matrices 1000×1000 en Python, qui demande 10^9 multiplications de nombres à virgule.

```
import torch, time

P = torch.rand(1000, 1000)
Q = torch.rand(1000, 1000)

t = time.perf_counter()
M = P @ Q
print(f'{time.perf_counter() - t:.02f}s')
```

affiche 0.01s.

Logiciels

Un ordinateur exécute des programmes pour produire des résultats qui sont ensuite saués dans des fichiers et/ou sont communiqués à des utilisateurs.

Plusieurs programmes peuvent être exécutés en même temps. Par exemple, celui qui gère l'affichage et la manipulation des fenêtres fonctionne en permanence.

Quand il démarre, l'ordinateur exécute un programme principal, son **système d'exploitation** qui permet essentiellement d'exécuter d'autres programmes.

Windows, Linux ou MacOS sont des systèmes d'exploitation.

La programmation consiste à décrire dans un ou plusieurs fichiers, sous une forme humainement intelligible, les opérations que l'ordinateur doit exécuter.

Le programmeur utilise un **éditeur de textes** interactif qui ressemble à un traitement de texte pour créer ces fichiers.

Il existe des programmes particuliers (“terminal”, “console”, “notebook”) qui permettent d'exécuter des opérations de manière interactive.

On différencie

- un **programme**, qui est une description que l'ordinateur comprend des opérations qu'il doit effectuer, de
- un **algorithme** qui est une décomposition formel en pas élémentaires destinée à être comprise par un humain.

Par exemple les procédures pour l'addition ou la multiplication en colonne sont des algorithmes.

Nous reviendrons sur cette distinction.

Par exemple, un algorithme pour calculer \sqrt{q} consiste à résoudre $x^2 - q = 0$ par dichotomie:

1. définir $a_0 = 0$, $b_0 = q + 1$
2. itérer tant que $b_n - a_n \geq \epsilon$:
 - $c_n = \frac{a_n + b_n}{2}$
 - Si $c_n^2 - q \geq 0$ alors $a_{n+1} = a_n$ et $b_{n+1} = c_n$,
 - sinon $a_{n+1} = c_n$ et $b_{n+1} = b_n$.

Cet algorithme écrit en C pourrait avoir la forme suivante:

```
float racine(float q, float epsilon) {
    float a, b, c;
    a = 0;
    b = q + 1;
    while(b - a >= epsilon) {
        c = (a + b) / 2;
        if (c * c - q >= 0) { b = c; }
        else { a = c; }
    }
    return c;
}
```


Développement d'un logiciel

1. Définir le problème.

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).
4. Tester le programme.

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).
4. Tester le programme.
5. (corriger, retourner ci-dessus...)

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).
4. Tester le programme.
5. (corriger, retourner ci-dessus...)
6. Intégrer le programme dans le système existant.

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).
4. Tester le programme.
5. (corriger, retourner ci-dessus...)
6. Intégrer le programme dans le système existant.
7. Tester l'intégration.

Développement d'un logiciel

1. Définir le problème.
2. Déterminer la méthode de résolution (**algorithmique**).
3. Écrire le programme correspondant (**programmation**).
4. Tester le programme.
5. (corriger, retourner ci-dessus...)
6. Intégrer le programme dans le système existant.
7. Tester l'intégration.
8. (corriger, retourner ci-dessus...)

Finalement, un principe fondamental de la programmation, qui explique la facilité avec laquelle des choses complexes peuvent être réalisées, est de réutiliser des “bouts de programmes” déjà écrits, *e.g.*

- calculs (algèbre linéaire, optimisation, résolution d'équations),
- opérations graphiques,
- cryptographie,
- serveurs web, communication réseau.

Finalement, un principe fondamental de la programmation, qui explique la facilité avec laquelle des choses complexes peuvent être réalisées, est de réutiliser des “bouts de programmes” déjà écrits, *e.g.*

- calculs (algèbre linéaire, optimisation, résolution d'équations),
- opérations graphiques,
- cryptographie,
- serveurs web, communication réseau.

Ces “bouts de programmes” sont regroupés dans des **librairies** qui n'ont d'utilité qu'utilisées dans un autre programme.

Fin