

# Les Universités du Tetal@b



Un algorithme :  
C'est quoi ?  
Ça sert à quoi ?  
Ça nous concerne ?

# 4000 ans d'utilisation connue

- ◆ -2000 Méthodes du fisc babylonien :
  - comptabilité sur tablettes (et non tableur 😊)
- ◆ -500 Euclide : PGCD
- ◆ -300 Aristote : définition de la logique
- ◆ 840 Dixit Algorismi
  - Muhammad Ibn Mūsā al-Khwarizmi
    - $3 \text{ sacs de riz} = 1 \text{ sac} + 50\text{Kg} \Rightarrow 1 \text{ sac} = ?$
- ◆ 1728 Falcon : métier à tisser à cartes perforées

# Quelques explications

« C'est une suite finie d'opérations (séquences) sur des objets (données) qui peut résoudre un problème ou obtenir un résultat. »

« Suite finie d'ordres précis qui sont exécutés automatiquement »

- ◆ Selon la nature des objets et des opérations on a :
  - une recette de cuisine
  - une méthode mathématique
  - une méthode utile dans la vie courante
  - une façon de communiquer
  - Etc ...
- ◆ Cela vise à systématiser la résolution d'un problème donné

# Bon à savoir

- ◆ Un algorithme s'applique à quelque chose :
  - Des objets
  - Des concepts
  - Des données
  - Des abstractions du monde réel
  
- ◆ Selon l'abstraction choisie un algorithme sera:
  - plus ou moins facile à comprendre
  - plus ou moins facile à traduire en code informatique
  - plus ou moins facile à corriger, modifier

# Concepts de 1840 à 2020

- ◆ 1840 Ada Lovelace / Charles Babbage
  - Notion de programme & itération
  - ➔ Invention du mot Algorithme
- ◆ 1940 Alan Mathison Turing
  - décodage chiffre Allemand avec ordinateur
  - 1ère définition de l'Intelligence Artificielle
- ◆ 2000 Gérard Berry
  - Formalisation & vérification de programme
  - L'Hyperpuissance de l'informatique :
  - Réflexion sur l'impact sociétal de l'algorithmique
- ◆ 2016 Cathy O'Neil
  - Apprentissage profond et megadata :
  - ➔ Aggravation des inégalités et menaces sur la démocratie

# Un algorithme courant, et d'actualité !

Ouvrir le robinet

Mouiller les mains

Les savonner pendant 30 secondes

Les rincer

Fermer le robinet

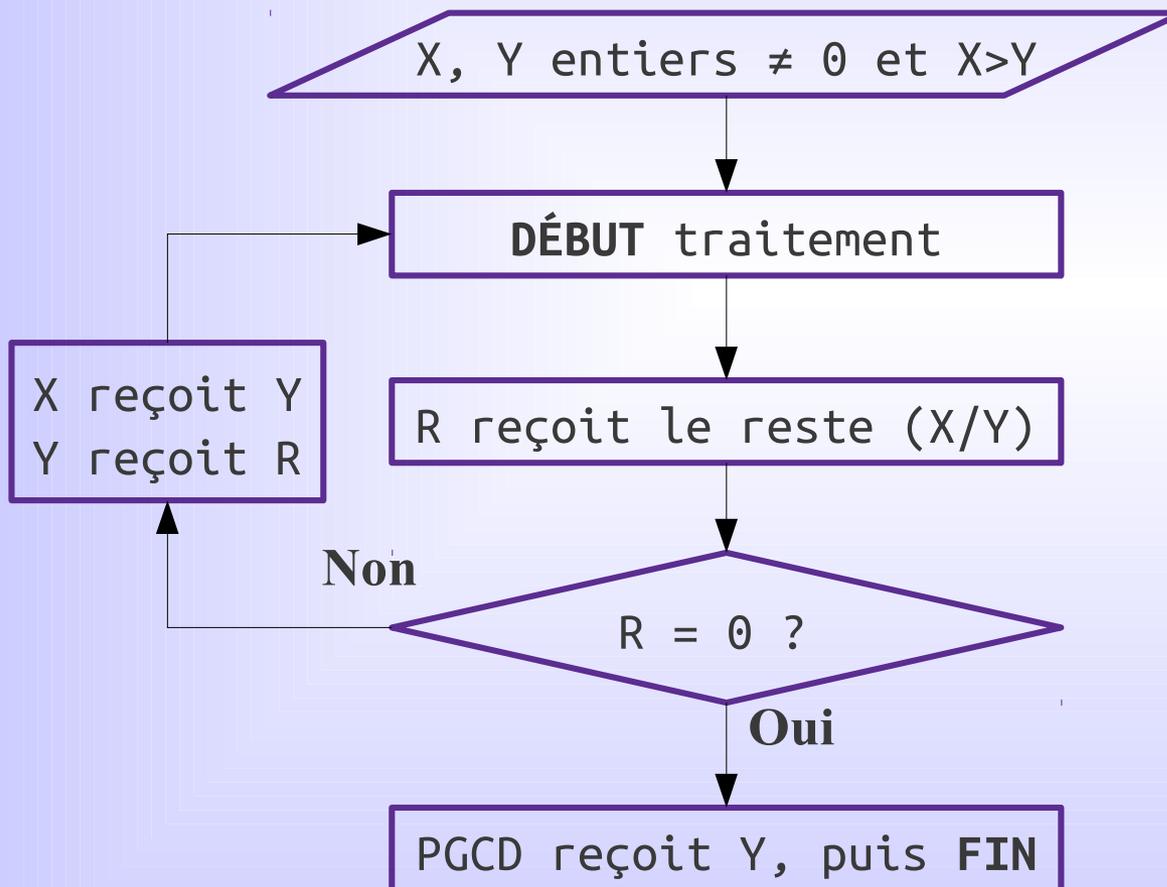
Essuyer les mains

Et ça ne coule pas de source !



# Exemple d'algorithme en math

- ◆ PGCD d'après Euclide, soit  $X$  et  $Y$  deux entiers :



- ◆ **DÉBUT**
- ◆  $R$  reçoit le reste de  $X$  divisé par  $Y$
- ◆ **Si**  $R = 0$ 
  - PGCD est  $Y$
  - **FIN.**
- ◆ **SINON**
  - Placer la valeur de  $Y$  dans  $X$
  - Placer la valeur de  $R$  dans  $Y$
  - **ALLER** au **DÉBUT**

# Comment s'écrit un algorithme

## ◆ Thèse de Church-Turing

- Un algorithme peut se décrire avec les structures :
  - La séquence
  - La sélection « Si »
  - La répétition « Tant que »

## ◆ Conséquence

- Tout langage décrivant un algorithme doit contenir ces 3 structures et peut se réduire à elles.

# Utilité des algorithmes

- ◆ On désire automatiser une procédure
  - Effectuer une action :
    - Changer d'étage dans un ascenseur
    - Afficher la page d'un site internet
    - Jouer un enregistrement sonore
    - ...
  - Résoudre un problème
    - Appliquer la méthode requise sur les données fournies

# Emploi des algorithmes

- ◆ Au début de l'informatique :
  - Calculer, Trier, Classer, Stocker.
- ◆ De nos jours :
  - Piloter des machines ou robots
  - Analyser des phénomènes
  - Faire des prévisions, apprendre à interpréter des données
  - Simuler des modèles de la réalité

# En automatique / informatique

- ◆ Il faudra inscrire l'algorithme dans la machine
- ◆ Nos machines sont électroniques
- ◆ Elles fonctionnent en langage binaire
- ◆ Toute information s'y écrit avec des 0 et des 1
- ◆ C'est pénible pour un humain normal !
- ◆ Invention des langages de programmation et des compilateurs à la fin des années 1950

# Les langages de programmation

- ◆ Un langage de programmation agit sur des données avec les instructions du langage
- ➔ Il y a des familles de langage, mais chaque langage est particulier
- ➔ Il existe plus d'un millier de langages
- ◆ Un compilateur est un programme. Il traduit le code source du langage en instructions binaires utilisables par la machine.

# Premiers langages

- ◆ 1954-56 John Backus : FORTRAN
- ◆ 1958-60 John Mac Carthy :  $\lambda$  calcul, LISP
- ◆ 1958-60 John Backus & Peter Naur : ALGOL
- ◆ 1959-60 Grace Hooper : bug, Cobol
- ◆ 1960 Donald Ervin Knuth : TAOCP, Tex
- ◆ 1970 Dennis Ritchie : langage C
- ◆ 1980 Niklaus Wirth : Pascal, Modula2, ADA

# Utilité des langages

- ◆ Un langage de programmation simplifie le codage d'un algorithme dans une machine
- ◆ Le plus souvent il comporte
  - Un lexique => des mots clés
  - Une syntaxe => façon d'agencer ces mots
- ◆ Le programmeur écrit dans ce « code source »
- ◆ Un programme « traduit » ce code source dans le code de la machine cible : Le compilateur
- ➔ Compilation du code source vers du code binaire

# En automatique / informatique

- ◆ Un langage propose des instructions de base :
  - Pour décrire des données et leurs types
  - Action d'entrée sortie de données
  - Action conditionnelle « Si »
  - Répétition d'une action « Tant que »
- ◆ Et des instructions de structuration
  - Sous-procédures, fonctions
  - Objets qui associent données et traitement pour modéliser des fonctionnements

# Quelques aphorismes ;-)

- ◆ L'algorithme prédit la continuité du passé
- ◆ Le code fait ce qu'on lui a dit, pas ce qu'on voudrait.
- ◆ Certains problèmes sont si complexes que nous n'avons pas d'algorithmes pour les résoudre.
- ◆ Il faut moins craindre l'IA que la bêtise naturelle
- ◆ En moyenne l'IA fait moins d'erreurs, mais les humains ne font jamais les erreurs qu'elle fait.
- ◆ Entre l'intelligence humaine et la connerie de la machine, il y a le logiciel. Gérard Berry

# Les algorithmes et nous et nous ...

- ◆ Ils sont massivement utilisés par
  - Les états
  - Les grandes sociétés
- ◆ Pour
  - Automatiser : administration, entreprises, industrie, armée ...
  - Accélérer les traitements : commerce, bourse, banque ...
  - Prévoir, influencer, contrôler nos comportement
  - Utiliser nos « clics » pour former des « IA » avec la méthode dénommée « apprentissage profond »

# Les algorithmes au 21ème siècle

- ◆ Ils sont dans tous nos objet numériques
- ◆ Ils participent à notre vie sociale / sexuelle
  - Réseaux, forums, rencontres, Porn, ...
- ◆ Ils assistent notre vie courante
- ◆ Ils tentent de reproduire notre intellect
- ◆ Ils aident les scientifiques
- ◆ Ils enrichissent massivement certaines sociétés
- ◆ Ils aident des médecins à sauver des vies
- ◆ Ils aident des états à asservir, voire tuer des personnes

# Data hyperphagie

- ◆ Des algorithmes collectent nos données ...
- ◆ Ceci produit un énorme volume de données
- ◆ Pour entraîner d'autres algorithmes !
- ◆ Difficile d'y échapper !
- ◆ Tout n'est pas analysable ou corrélable
- ◆ Mais pour combien de temps encore ?

# Algorithmopathie

- ◆ « Weapons of math destruction » de Cathy O'Neil démontre l'impact social envers les plus défavorisés.
- ◆ À son insu, chacun est en situation d'asymétrie informationnelle vis à vis des opérateurs du système.
- ◆ Ils ont accès à une part de notre sphère intime.
- ◆ La technologie progresse :
  - Demain meilleure analyse des données d'hier
  - Quel sera alors le pouvoir des détenteurs ?
  - Allons-nous en souffrir ? Un peu, beaucoup ?

# Soins palliatifs

- ◆ Surfez couvert : TOR, VPN
- ◆ Créez du bruit, offuscation
  - TrackMeNot
  - Un compte mail associé à un type de navigation
  - Plusieurs SIM
  - Plus extrême : un terminal / PC par activité
- ◆ Préférez des alternatives éthiques, libres
  - Framasoft
  - Mastodon
  - Diaspora

# Consulter / Lire

## ◆ Guide d'auto défense numérique

- <https://guide.boum.org>

## ◆ <https://cfeditions.com/public>

- « Surveillance:// » de Tristan Nitot

- « Obfuscation » de Finn Brunton & Helen Nissenbaum

- « L'appétit des géants » de Olivier Ertzscheid

## ◆ SITOthèque

- <https://mathbabe.org>

- [https://www.affordance.info/mon\\_weblog/intelligence-artificielle/](https://www.affordance.info/mon_weblog/intelligence-artificielle/)

- [https://www.affordance.info/mon\\_weblog/big-data/](https://www.affordance.info/mon_weblog/big-data/)

- <https://www.franceculture.fr/emissions/linvite-des-matins/les-algorithmes-regissent-ils-nos-vies>

# Merci d'être venu, car :

« Pour comprendre la nature essentielle d'un objet technique, la relation d'usage n'est pas la plus pertinente car la répétition des gestes estompe la conscience que l'on peut avoir des structures et des fonctionnements internes de la machine. »

Jacques Munier à propos de l'oeuvre de  
Gilbert Simondon : Du mode d'existence des  
objets techniques (Aubier)

# Quelques pensées

"WHERE IS THE LIFE WE HAVE LOST IN LIVING?

WHERE IS THE WISDOM WE HAVE LOST IN KNOWLEDGE?

WHERE IS THE KNOWLEDGE WE HAVE LOST IN INFORMATION?"

T.S. ELIOT, IN CHORUSES FROM THE ROCK (1934)

WHERE IS THE INFORMATION WE'VE LOST IN GOOGLE ?

ERTZSCHEID OLIVIER,

WHERE ARE THE (TRUE) RELATIONSHIPS WE'VE LOST IN FACEBOOK ?