

# Introduction à l'algorithmique et à Java

Licence 1 MASS

Introduction à Java et à l'algorithmique

Olivier Dalle

dalle@unice.fr

`<deptinfo.unice.fr/~dalle/>`

NB: Ce cours a été mis au point par mon collègue Sébastien Vérel, actuellement en congé de recherche.

Équipe Mascotte, commune I3S - CNRS/UNS & INRIA  
Sophia Antipolis

19 janvier 2009

## Objectifs de la séance 1

- 1 Connaître des algorithmes “historiques”
- 2 Connaître une définition d'algorithme
- 3 Connaitre l'environnement Processing
- 4 Editer et commenter un code java avec processing
- 5 Savoir afficher un texte
- 6 Savoir afficher des formes graphiques simples

Question principale du jour :

Qu'est-ce qu'un algorithme ?

# Plan

- 1 Exemples d'algorithmes de l'histoire et du quotidien
- 2 Une très courte histoire de l'algorithmique
- 3 Une courte histoire de l'informatique et des ordinateurs
- 4 Définitions d'Algorithme
- 5 Le langage Java

# Algorithme d'Euclide (*éléments*, VII, -325/-265 av. JC)

## Problème

Trouver une "unité de mesure" commune pour deux longueurs de segments *i.e.* trouver le pgcd deux nombres entiers.

**Algorithme** PGCD( $a, b$  : entier) : entier

**début**

si  $b = 0$  alors

PGCD =  $a$

sinon

$c \leftarrow$  reste de la division de  $a$  par  $b$

PGCD = PGCD( $b, c$ )

fin si

**fin**

## Exécution de l'algorithme PGCD(72, 34)

Pour  $a = 72$  et  $b = 34$

1. PGCD(72, 34)

3.  $b \neq 0$

6.  $c = 4$

7. PGCD(34, 4)

3.  $b \neq 0$

6.  $c = 2$

7. PGCD(4, 2)

3.  $b \neq 0$

6.  $c = 0$

7. PGCD(2, 0)

3.  $b = 0$

**Algorithme** PGCD( $a, b$  : entier) : entier

**début**

**si**  $b = 0$  **alors**

PGCD =  $a$

**sinon**

$c \leftarrow$  reste de la division de  $a$  par  $b$

PGCD = PGCD( $b, c$ )

**fin si**

**fin**

# Crible d'Erastothène (III<sup>e</sup> avant J.-C.)

## Problème

Déterminer les nombres premiers inférieur à une borne supérieure.

**Algorithme** Erastothene( $N$  : entier) : tableau d'entiers

**début**

Ecrire sous forme de tableau les nombres entiers  
compris entre 2 et  $N$ .

**tant que** carré du plus petit nombre non rayé et  
non marqué est plus petit que  $N$  **faire**

Marquer le plus petit nombre du tableau non rayé et non marqué

Rayer du tableau tous les multiples de ce nombre

**fin tant que**

nombre premiers inférieurs à  $N$  : nombres marqués ou non rayés

**fin**

## Crible d'Erastothène (III<sup>e</sup> avant J.-C.)

### Remarques

Possibilité d'amélioration de l'algorithme :

→ ne pas écrire les multiples de 2 et les multiples de 5.

- Avantage :

plus rapide à écrire au début parce que moins de place utilisé sur une feuille

- Inconvénient :

plus long à rayer : il faut réfléchir plus longtemps pour rayer les nombres adéquates !

Version moderne de l'algorithme :

- Crible d'Atkin (1999)

Référence : A.O.L. Atkin, D.J. Bernstein, Prime sieves using binary quadratic forms, Math. Comp. 73 (1999), 1023-1030.

# Multiplication égyptienne

## Problème

Calculer le produit de 2 entiers

**Algorithme** MultiplicationEgyptienne( $a, b$  : entier) : : entier

### début

Ecrire la table des puissances de 2 inférieures ou égales au nombre  $a$

Ecrire la table des doubles du nombre  $b$

**tant que** le nombre  $a$  n'est pas nul **faire**

    Cocher la plus grande puissance de  $2 \leq a$

    Soustraire cette puissance de 2 du nombre  $a$

**fin tant que**

additionner les doubles du nombre  $b$  correspond aux puissances de 2  
cochés précédemment

produit de  $a$  par  $b$  : somme calculée ci-dessus

### fin



# Tour de magie!

Choisissez un nombre entre 1 et 15.

# Tour de magie!

Est-il dans ce tableau ?

9	14	10
15	8	13
12	11	

# Tour de magie!

Est-il dans ce tableau ?

15	3	11
13	9	5
7	1	

# Tour de magie!

Est-il dans ce tableau ?

6	14	13
12	4	5
15	7	

# Tour de magie!

Est-il dans ce tableau ?

2	3	6
7	10	11
14	15	

## Tour de magie!

Connaissant vos réponses, je peux trouver quel nombre vous avez choisi.

# Recherche d'un mot dans un dictionnaire

## Itérativement

**Algorithme** `rechercher(cible : mot)` : liste de mots  
**début**

Lire premier mot du dictionnaire

**tant que** mot lu n'est pas le mot cible **faire**

Lire le mot suivant

**fin tant que**

liste de mots : définition du mot lu

**fin**

On trouve le mot correct mais l'algorithme n'est pas très efficace...

- en moyenne,  $N/2$  mots lus avec  $N$  la taille du dictionnaire
- Il faut utiliser l'ordre lexicographique du dictionnaire
- et la méthode dichotomique (diviser pour régner)

# Recherche d'un mot dans un dictionnaire par dichotomie

**Algorithme** rechercheDicho(*cible* : mot) : liste de mots  
**début**

premier ← premier mot du dictionnaire

dernier ← dernier mot du dictionnaire

Lire le mot médian entre premier et dernier

**tant que** mot lu n'est pas le mot cible **faire**

**si** mot cible est avant le mot lu **alors**

        dernier ← mot lu

**sinon**

        premier ← mot lu

**fin si**

    Lire le mot médian entre premier et dernier

**fin tant que**

liste de mots : définition du mot lu

**fin**

- en moyenne,  $\log_2(N)$  mots lus avec N la taille du dictionnaire



# Indice de masse corporelle

## Problème

Donner un degré de corpulence basé sur l'indice de masse corporelle

Algorithme `degreMasseCorporelle`( $T$  : nombre réel,  $m$  : nombre réel) :

début

$i \leftarrow m/T^2$

si  $i < 20$  alors

    écrire( "poids inférieur à la normale")

sinon

    si  $i < 25$  alors

        écrire( "poids normal")

    sinon

        si  $i < 30$  alors

            écrire( "surcharge pondérale")

        sinon

            si  $i < 40$  alors

                écrire( "adiposité")

            sinon

                écrire( "obésité")

        fin si

    fin si

fin si

fin si

fin

# Bouilloire

## Problème

Concevoir une bouilloire pour faire bouillir de l'eau

je suppose qu'une bouilloire (électrique) peut :

- connaître la température de l'eau  $T$
- connaître et modifier la position de son interrupteur
- chauffer

**Algorithme** bouilloire( ) :

**début**

**tant que** la prise est branchée **faire**

**tant que** interrupteur est ON **et**  $T < 100$  **faire**

Chauffer

**fin tant que**

positionner interruption sur OFF

**fin tant que**

**fin**

# Recettes de cuisine

## Problème

### Convevoir un (bon) gâteau

#### *Préparation :*

*Ingrédients* (pour 6 personnes) :

- 1 pâte brisée
- 150 g de sucre
- 100 g de beurre fondu
- 3 oeufs
- le jus de deux citrons

- . Préchauffer le four à 200°C.
- . Abaisser la pâte brisée.
- . Battre les oeufs avec le sucre en poudre jusqu'à l'obtention d'un mélange moussieux.
- . Ajouter le jus de citron.
- . Ajouter le beurre fondu.
- . Enfourner et laisser cuire environ 30 mn.
- . La préparation doit dorer.

# Vos algorithmes quotidiens

En connaissez-vous d'autres ?

## Vieille histoire

- Descriptions exhaustives d'algorithmes :  
dès l'époque des Babyloniens (de -2000 av JC à -300 av JC),
- Pour des calculs concernant le commerce et les impôts.

## Al-Khwarizmi (né vers 780 - mort vers 850)

**algorithme** : vient du nom du mathématicien perse du IX<sup>ème</sup> siècle

Abu Djafar Muhammad ibn Musa al-Khwarizmi

- mot algorithme : à l'origine uniquement aux règles d'arithmétique utilisant les chiffres indo-arabes numéraux
- Traduction en latin européen du nom Al-Khwarizmi's en "algorithme" au XVIII<sup>ème</sup> siècle.
- Evolution pour inclure toutes les procédures définies pour résoudre un problème ou accomplir une tâche.

## Quelques dates jusqu'en 1946

principale source : <http://histoire.info.online.fr/prehistoire.html>

- -3000 : Période de l'empereur Chinois Fou-Hi dont le symbole magique, l'octogone à trigramme contient les 8 premiers nombres représentés sous forme binaire par des traits interrompus ou non : 000 001 010 011 etc...
- -500 : Apparition au Moyen Orient du premier "outil" de calcul : l'abaque et le boulier.
- 1580 : John NAPIER invente les logarithmes.
- 1623 : Wilhelm Schickard invente ce qu'il appelle une horloge calculante. Elle calculait mécaniquement grâce à des roues dentées et pouvait réaliser additions, soustractions, multiplications et mémorisation des résultats intermédiaires. La machine a rapidement sombré dans l'oubli car son inventeur habitait en Allemagne du Sud dans une région ravagée par la guerre de 30 ans.

## Quelques dates

- 1632 : L'Anglais Oughtred invente la Règle à calcul.
- 1642 : Pascal met au point, pour aider son père collecteur des impôts à Rouen, la Pascaline qui pouvait traiter les additions et les soustractions.
- 1679 : Leibnitz découvre et met au point une arithmétique binaire (et analyse les octogrammes de Fou-Hi). Il invente aussi en 1694 une machine à calculer dérivée de la Pascaline mais capable de traiter les multiplications et divisions.



## Quelques dates

- 1833 : Babbage : machine à différences puis une machine analytique qui contient les concepts de ce que sera l'ordinateur moderne : unité de calcul, mémoire, registre et entrée des données par carte perforée.
- 1836 - 1838 : Les Anglais Edward Davy, William Looke et Charles Wheastone vont inventer et mettre au point le télégraphe. Le peintre Américain Samuel Morse invente le code qui porte son nom
- 1840 : Collaboratrice de Babbage, Ada Lovelace, mathématicienne, définit le principe des itérations successives dans l'exécution d'une opération.  
Créatrice du mot algorithme.

## Quelques dates

- 1876 : L'Américain Graham Bell invente le téléphone et fonde la compagnie Bell Telephone Company.
- 1884 : Herman Hollerith crée une tabulatrice à cartes perforées (inspirée des métiers à tisser de Jacquard) pour réaliser le recensement Américain de 1890.  
première machine à traiter l'information.
- 1924 : La firme créée par Herman Hollerith en 1896, Tabulating Machine Corporation, est renommée en International Business Machine ou IBM.
- 1935 : IBM commercialise l'IBM 601, un calculateur à relais utilisant des cartes perforées capable de réaliser une multiplication en une seconde.  
Il en sera vendu 1500 exemplaires essentiellement pour les marchés scientifiques et comptables.

## Quelques dates

- 1937 : **Alan Turing** publie un document sur les nombres calculables. Il résolvait des problèmes mathématiques en utilisant une sorte d'ordinateur logique très simple appelé depuis Machine de Turing.
- 1938 : Thèse de Shannon qui le premier fait le parallèle entre les circuits électriques et l'algèbre Booléenne. définition du chiffre binaire : bit (Binary digiT).
- 1938 : Création du Versuchmodell 1 ou Z1 par Konrad Zuse. Il le met au point dans le salon de ses parents à Berlin ! Il s'agit d'un ordinateur binaire programmable mais mécanique. Il ne fonctionna jamais vraiment correctement.

## Quelques dates

- 1939 : Konrad Zuse et un de ses amis Helmut Schreyer : deuxième ordinateur, le Z2  
remplacement partie des pièces mécaniques du Z1 par relais électromécaniques de téléphone rachetés d'occasion.
- 1940 : Pour décrypter les messages de l'armée Allemande, les Anglais mettent au point sur le site de Bletchley Park les calculateurs Robinson et Colossus sous la direction du mathématicien Alan Turing.
- 1941 : Création du calculateur binaire ABC par John Atanasoff et Clifford Berry. La machine utilise des lampes et comporte une mémoire et des circuits logiques.  
mémoire : 60 mots de 50 bits, fréquence 60 Hz, 1 addition / s.  
Premier vrai ordinateur ? (programme n'est pas en mémoire)

## Quelques dates

- 1941 : Konrad Zuse, mobilisé dans les usines d'aviation Henschel : Z3 (programme enregistré).  
premier véritable ordinateur.  
2600 relais, console pour l'opérateur et d'un lecteur de bandes contenant les instructions à exécuter. stockage : 64 nombres de 22 bits.  
4 additions par seconde et 1 multiplication en 4 secondes.
- 1943 : Création du ASCC Mark I (Automatic Sequence-Controlled Calculator Mark I) à Harvard par Howard Aiken et son équipe (avec le soutien d'IBM).  
3000 relais, 800 km de cables  
3 opérations sur 23 chiffres par seconde.  
Pas de sous programme, pas de branchement.

## Quelques dates

- 1945 : Un insecte coincé dans les circuits bloque le fonctionnement du calculateur Mark I. La mathématicienne Grace Murray Hopper invente le mot BUG...
- 1945 : **John Von Neuman** : description d'ordinateur à programme enregistré  
EDVAC (Electronic Discrete Variable Automatic Computer).
- 1946 : Création de l'ENIAC (Electronic Numerical Integrator and Computer) par P. Eckert et J. Mauchly. La programmation de ce calculateur s'effectue en recablant entre eux, ses différents éléments.  
19000 tubes, 30 tonnes, 72 m<sup>2</sup> et consomme 140 kilowatts.  
Horloge : 100 KHz. Vitesse : environ 330 multiplications par seconde.

## Génération d'ordinateur

- Décembre 1947 : Invention du transistor par William Bradford Shockley, Walter H. Brattain et John Bardeen dans les laboratoires de Bell Telephone.  
→ Deuxième génération d'ordinateur
- 1958 : Démonstration du premier circuit intégré crée par Texas Instruments.  
→ troisième génération d'ordinateur
- 1965 : Gordon Moore écrit que la complexité des circuits intégrés doublera tous les ans : "Loi de Moore".
- 1971 : l'ère du microprocesseur (INTEL)  
→ Quatrième génération

## Définition d'Ordinateur

En 1955, "ordinateur" : traduction J. Perret de " electronic data processing machine " ( machine électronique de traitement des données ).

→ ce que beaucoup de langues nomme "calculateur" (computer)

### Définition Académie française de 1967

Une machine automatique qui permet d'effectuer, dans le cadre de programmes de structure pré-établis, des ensembles d'opérations arithmétiques et logiques à des fins scientifiques, administratives ou comptables.



## Définition informatique

En 1962, Philippe Dreyfus employa le mot informatique pour définir le *traitement automatique de l'information*.

Maintenant ce mot est employé dans beaucoup de langues (hors anglaise)

→ Informatique : composé des 2 mots *information* et *automatique*.

### Définition Académie française de 1967 (officialisation)

Science du traitement rationnel, notamment à l'aide de machines automatiques, de l'information, considérée comme le support de connaissances dans les domaines scientifique, économique et social.

## Retour sur les exemples

Quels sont les points communs entre tous ces “algorithmes” ?

- algorithme d'Euclide
- algorithme d'Erastothène
- multiplication égyptienne
- degré de masse corporelle
- recette de cuisine

# Caractéristiques d'algorithme

## Résolution d'un problème

- algorithme d'Euclide :  
Trouver une unité commune à deux longueurs
- algorithme d'Erastothène :  
Trouver les nombres premiers plus petit que N
- Multiplication égyptienne :  
Calculer le produit de deux nombres
- degré de masse corporelle :  
Donner le degré de corpulence basé sur l'IMC
- recette de cuisine :  
Confectionner le gâteaux de vos rêves

# Caractéristiques d'algorithme

## Résolution de manière opératoire

- algorithme d'Euclide :  
“Calculer le quotient de  $a$  par  $b$ ”
- algorithme d'Erastothène :  
“rayer les multiples de ...”
- Multiplication égyptienne :  
“ $a \leftarrow E[a/2]$ ”
- degré de masse corporelle :  
“Si  $i < 25$  alors ...”
- recette de cuisine :  
“Mélanger les oeufs avec ...”

# Caractéristiques d'algorithme

Énoncé dans un langage précis

Langage le plus formel : sémantique opérationnelle, axiomatique, ...

- algorithme d'Euclide :  
"quotient"
- algorithme d'Erastothène :  
"rayer les multiples de ..."
- Multiplication égyptienne :  
" $a \leftarrow E[a/2]$ "
- degré de masse corporelle :  
"Si  $i < 25$  alors ..."
- recette de cuisine :  
"Mélanger les oeufs avec ..."

# Caractéristiques d'algorithme

Données / informations en entrée

- algorithme d'Euclide :  
les deux nombres  $a$  et  $b$
- algorithme d'Erastothène :  
les nombres entiers
- Multiplication égyptienne :  
les deux opérands
- degré de masse corporelle :  
le poids et la taille
- recette de cuisine :  
Les ingrédients

# Caractéristiques d'algorithme

Données / informations en sortie

- algorithme d'Euclide :  
le PGCD
- algorithme d'Erastothène :  
les nombres premiers plus petits que  $N$
- Multiplication égyptienne :  
le produit des nombres
- degré de masse corporelle :  
un classement
- recette de cuisine :  
un gâteau au citron !

# Caractéristiques d'algorithme

S'arrête en temps fini

- algorithme d'Euclide :  
au plus  $\max(a, b)$  ?
- algorithme d'Erastothène :  
au plus  $\sqrt{NN}$  ?
- Multiplication égyptienne :  
au plus  $\log(a)$  ?
- degré de masse corporelle :  
5 ou 6 opérations
- recette de cuisine :  
2h...



## Tentatives de définitions

### définition un peu courte

Un algorithme énonce une résolution sous la forme d'une série d'opérations à effectuer.

### définition (dictionnaire canadien)

Formule ou ensemble d'étapes qu'on applique pour résoudre un problème en particulier. Un algorithme doit avoir un ensemble de règles sans ambiguïté et un point limite bien défini

## Tentatives de définitions

### définition (Wikipedia)

Un algorithme est un moyen pour un humain de présenter la résolution par calcul d'un problème à une autre personne physique (un autre humain) ou virtuelle (un calculateur). En effet, un algorithme est un énoncé dans un langage bien défini d'une suite d'opérations permettant de résoudre par calcul un problème.

### définition satisfaisante

Un algorithme est un ensemble fini d'instructions correctement définis qui a pour but la résolution d'une tâche à partir d'un état initial et se terminant sur un état final bien défini.

## Tentatives de définitions, une dernière

### Définition ("complexité et algorithmique avancée", Ivan Lavallé)

Un algorithme y est vu comme une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat, et cela indépendamment des données.

## Vers une définition formelle

→ Il n'existe pas une définition formelle admise par tous.

Premières formalisations :

- Alan Turing (1936) : un algorithme est ce qui exécutable à partir d'une machine de Turing.
- Alonzo Church (lambda calculus) : tout calcul est réalisable à l'aide d'un algorithme exécuté par un ordinateur (avec suffisamment de temps et d'espace)

## ApplicationSSSS

- Cryptographie (paiement sécurisé, téléphone portable, connections internet, etc)
- Compression d'images (dvd, mpeg4, etc), de sons (mp3, etc)
- internet (moteur de recherche,...)
- gps (tracer de route, ...)
- organiser son temps, son travail
- jeux
- problèmes industriels (conception automatisé ou assister...)
- la cuisine facile !
- résolution d'équations (communications numériques, ...)
- intelligence artificielle (optimisation, apprentissage automatique, ...)

## Grands domaines d'applications

- Impôts, l'administration public, etc.
- Commerce, la finance, l'administration privée, etc.
- Calcul scientifique
- Loisirs : musique, vidéo, jeux, etc.
- Communication : réseaux internet, mobile, communication satellite, etc.

## Langages de programmation

Un algorithme est un énoncé générale qui peut s'exprimer dans un grand nombre de langages de programmation

Lorsqu'un algorithme est exprimé dans un certain langage particulier,

on dit que l'on **implémente**  
ou que l'on **code** un algorithme dans ce langage.

## Classification des langages de programmation

On distingue plusieurs type de langage de programmation selon leurs propriétés :

- langages impératifs :  
fortran, basic, pascal, bash, C, maple, php, javascript
- langage fonctionnel :  
lisp, scheme, caml
- langage orienté objets :  
eiffel, C++, java, php5
- ....

Les langages actuels tendent à combiner toutes les approches de programmation possible.



# Langage Java et système Processing

- Langage Java : langage orienté Objet
- Système Processing :
  - simplification de Java,
  - interface entre utilisateur et java,
  - Environnement de programmation

# Système Processing

- Créé au MIT (Boston, USA) pour introduire les concepts de base de la programmation dans le contexte des arts visuels (artistes, Web).
- Idée : immersion rapide dans les programmes graphiques, l'Art Numérique et la simulation d'expériences physiques!...
- Environnement graphique mis en avant

## Installation, configuration

- Un cours assez complet orienté "non informaticien" :  
<http://www.ecole-art-aix.fr/rubrique81.html>
- Le cours 02 concernant l'installation

## Affichage de texte

- Sans retour à la ligne :  
`print("super genial megacool");`
- Avec retour à la ligne :  
`println("yes we can !");`

**Attention !** les instructions se terminent par un point virgule ;

## Ecrire un commentaire

- Les commentaires sont des lignes non interprétées par l'ordinateur
- Importance capitale pour la lisibilité de votre programme
- Importance capitale pour communiquer comment fonctionne votre programme
- Je **Veux** et j'**Exige** des commentaires !

```
// dessin du tableau
```

```
// fin du programme
```

## Graphisme : couleur de fond

```
background(r, v, b) ;
```

où  $r$ ,  $v$  et  $b$  sont des entiers entre 0 et 255 donnant le niveau de rouge, vert et bleu.

## Graphisme : Dimension de la fenêtre

- pixel (picture element) : petit rectangle élémentaire assimilé à un "point"
- un écran 1024x768 : signifie qu'il y a 1024 pixels en largeur et 768 pixels en hauteur

Réglage de la taille de la fenêtre graphique :

```
size(largeur, hauteur) ;
```

## Graphisme en vrac

```
point(x, y) ;
```

```
line(x1, y1, x2, y2) ;
```

```
rect(x, y, largeur, hauteur) ;
```

```
ellipse(x, y, largeur, hauteur) ;
```



## Objectifs de la séance 1

- 1 Connaître des algorithmes “historiques”
- 2 Connaître une définition d'algorithme
- 3 Connaitre l'environnement Processing
- 4 Editer et commenter un code java avec processing
- 5 Savoir afficher un texte
- 6 Savoir afficher des formes graphiques simples

Question principale du jour :

Qu'est-ce qu'un algorithme ?

## Travail pour la semaine prochaine

- Combien d'étagères sont consacrées à la programmation en JAVA à la BU ?
- Choisir un livre qui vous semble en rapport avec cet enseignement
- Trouver au moins un site web en rapport avec l'enseignement
- Installer Processing sur votre ordinateur si vous en avez un