Introduction à l'informatique et la programmation
IF1 2010-2011

Matthieu Picantin

LIAFA UMR 7089
CNRS & Université Paris 7 Denis Diderot

16/09/2010

Organisation générale

- Cours le jeudi de 14h30 à 16h30 en amphi 2A
  - groupe α : 16/09 14/10 28/10 25/11 09/12
  - groupe β : 23/09 07/10 04/11 18/11 16/12
- Cours-TD
  - 2 heures par semaine
  - début la semaine du 20/09
- TP
  - 2 heures par semaine (TP1)
    - ⋆ début sem. du 20/09
  - 2 heures une semaine sur deux (TP2)
    - ⋆ début sem. du 20/09 pour tous sauf MASS2, MATH3 & PHY3
    - ⋆ début sem. du 27/09 pour MASS2, MATH3 & PHY3
- Tutorat
  - du lundi au jeudi de 12h30 à 14h30 au SCRIPT

Tests et examens

- Td : Résultat des tests effectués en TD
- Tp : Résultat des tests effectués en TP
- E0 : Partiel (samedi 20 novembre)
- E1 : Examen en janvier
- E2 : Examen de juin

Notes finales

- Contrôle continu : Cc = (Td + Tp) / 2
- Note d'écrit : Ne = max (E1, (E0 + E1) / 2)
- Note finale janvier : (3Ne + Cc) / 4
- Note finale juin : max (E2, (3E2 + Cc) / 4)

Notes planchers

- INFO, MI, CPI & MASS-Info-Linguistique : note plancher 07/20
- MATH, PHY & autres MASS : note plancher 00/20

Programme

- notion d'algorithme et son expression en langue naturelle
- langages de programmation (styles de programmation, compilation/interprétation, code binaire ou code intermédiaire)
- variables, identificateurs, concept de type
- affectation, expressions (arithmétiques, booléennes) et leur évaluation
- structures de contrôle : sélections, itérations
- tableaux à une ou plusieurs dimensions et imbrications de boucles
- concept de fonction et transmission de paramètres
- rapide introduction au concept d'objet

Objectifs

- comprendre un certain nombre des concepts généraux des machines et de la programmation
- réaliser le codage effectif, la compilation et l'exécution d'algorithmes simples dans un environnement de type Unix

Bibliographie

- Concepts fondamentaux de l'informatique
  Alfred Aho & Jeffrey Ullman
  (une dizaine d'exemplaires à la bu 004 AHO)
- The Java Tutorial (fourth edition),
  Mary Campione, Kathy Walrath & Alison Hulm
  `http://java.sun.com/docs/books/tutorial/`
- Thinking in Java,
  Bruce Eckel
  `http://www.mindview.net/Books/TIJ/` ou
  `http://penserenjava.free.fr/`

Informations relatives au cours

`http://www.liafa.jussieu.fr/~picantin/IF1/`

Informations sur la classe `Deug`

`http://www.liafa.jussieu.fr/~yunes/deug/`

Qu'est ce que l'algorithmique ?

- Avez-vous déjà ouvert un livre de cuisine pour concocter une bonne ouiche lorraine ?
- Avez-vous déjà déchiffré une notice de montage d'une armoire suédoise ?
  - **Si oui, sans le savoir, vous avez déjà exécuté des *algorithmes* !**

- Avez-vous déjà indiqué son chemin à un touriste égaré ?
- Avez-vous déjà écrit une lettre anonyme précisant comment procéder à une remise de rançon ?
  - **Si oui, vous avez déjà fabriqué—et fait exécuter—des algorithmes !**

L'art d'organiser un calcul complexe
en *partant d'opérations simples*

une suite d'instructions compréhensibles par celui qui devra l'exécuter

Qu'est ce que l'algorithmique ?

L'art d'organiser un calcul complexe
en partant d'opérations simples

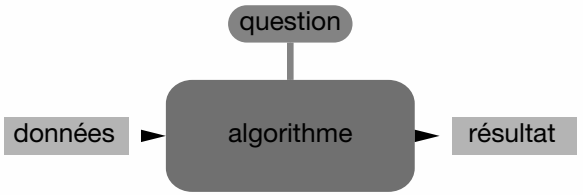
Étymologie

Le mot *algorithme* vient du nom du mathématicien et astronome perse Al Khuwarizmi, qui écrivit au IX^{e} siècle un traité donnant la première solution systématique des équations quadratiques

Quelques grands

Euclide, Archimède, Brahmagupta, Al Khuwarizmi, Fibonacci Hilbert, Turing, Church, Gödel, Von Neumann, Knuth, Karp,...

L'art d'organiser un calcul complexe en partant d'opérations simples



Instructions

- affectation
- lecture/écriture
- tests
- boucles

Méthodes

- approche incrémentale
- principe du diviser pour régner
- technique gloutonne
- exploitation de l'aléa

- le type de machine
- le temps de calcul
- la taille mémoire
- la consommation d'énergie
- le type de résultat

images
texture éclairage
fusion mise 3D
contours segmentation
corrections optiques

géométrie
trajectoires
déformations
tomographie
surfaces volumes

réseaux
gestion de trafic
diffusion
protocoles
codage

mots, textes
grammaires, automates
croisement
tri, recherche
classement

nombres
éléments finis
matrices prog. linéaire
premiers cryptage
4 opérations

optimisation
emploi du temps
ordonnancement
circulation
routage

- le type de machine
- le temps de calcul
- la taille mémoire
- la consommation d'énergie
- le type de résultat

Objets
- bits, entiers, flottants
- graphes, matrices
- mots, images

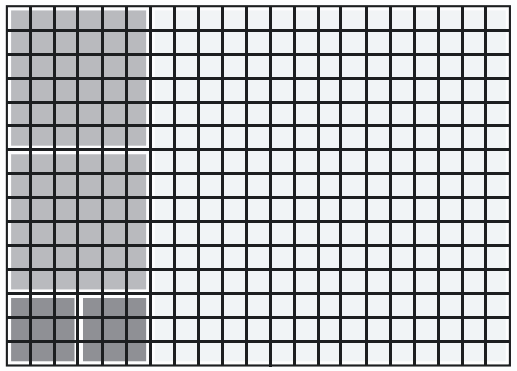
Structures de données
- piles, files
- listes, tableaux
- arbres, tas

Structures de contrôle
- séquence, boucle
- récursion
- parallélisme synchrone
- parallélisme asynchrone

- algorithmes de tri
  - tri par insertion
  - tri par dichotomie-fusion
  - tri rapide
- algorithmes pour arithmétique
  - addition
  - pgcd et division euclidienne
  - fonctions de Ackermann
- algorithmes géométriques
  - enveloppe convexe
  - diagramme de Voronoï
  - triangulation de Delaunay
- algorithmes stochastiques
  - calcul de π
  - chifoumi
- modèles abstraits
  - machine de Turing
  - Jeu de la Vie

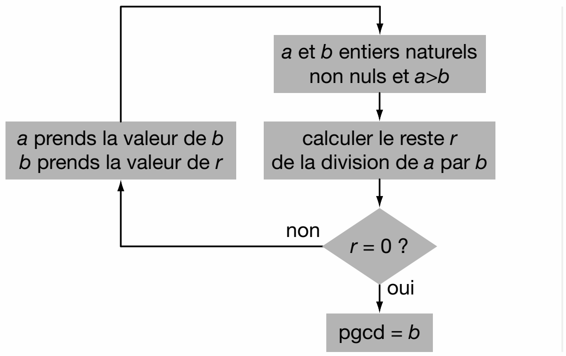
L'algorithme d'Euclide est un algorithme permettant de déterminer le plus grand commun diviseur (pgcd) de deux entiers dont on ne connaît pas la factorisation.
Il est déjà décrit dans le livre VII des Éléments d'Euclide.

Calcul géométrique du pgcd de 15 et 21 : 3



L'algorithme d'Euclide est un algorithme permettant de déterminer le plus grand commun diviseur (pgcd) de deux entiers dont on ne connaît pas la factorisation.
Il est déjà décrit dans le livre VII des Éléments d'Euclide.

Organigramme



Définition

L'algorithme d'Euclide est un algorithme permettant de déterminer le plus grand commun diviseur (pgcd) de deux entiers dont on ne connaît pas la factorisation.
Il est déjà décrit dans le livre VII des Éléments d'Euclide.

Pseudo-code

soient a et b deux entiers
si $b = 0$
    alors le pgcd est a
    sinon le pgcd est celui de b et r,
        où r est le reste de la division entière de a par b

12	17	10	23	33	77	83	11	39	43	14	18	15	31	91	24
12	17	10	23	33	77	83	11	39	43	14	18	15	31	91	24
10	12	17	23	33	77	83	11	39	43	14	18	15	31	91	24
10	12	17	23	33	77	83	11	39	43	14	18	15	31	91	24
10	12	17	23	33	77	83	11	39	43	14	18	15	31	91	24
10	12	17	23	33	77	83	11	39	43	14	18	15	31	91	24
10	12	17	23	33	77	83	11	39	43	14	18	15	31	91	24
10	11	12	17	23	33	77	83	39	43	14	18	15	31	91	24
10	11	12	17	23	33	39	77	83	43	14	18	15	31	91	24
10	11	12	17	23	33	39	43	77	83	14	18	15	31	91	24
10	11	12	14	17	23	33	39	43	77	83	18	15	31	91	24
10	11	12	14	17	18	23	33	39	43	77	83	15	31	91	24
10	11	12	14	15	17	18	23	33	39	43	77	83	31	91	24
10	11	12	14	15	17	18	23	31	33	39	43	77	83	91	24
10	11	12	14	15	17	18	23	31	33	39	43	77	83	91	24
10	11	12	14	15	17	18	23	24	31	33	39	43	77	83	91

Exemple de tri par dichotomie-fusion

12 17 10 23 33 77 83 11 39 43 14 18 15 31 91 24

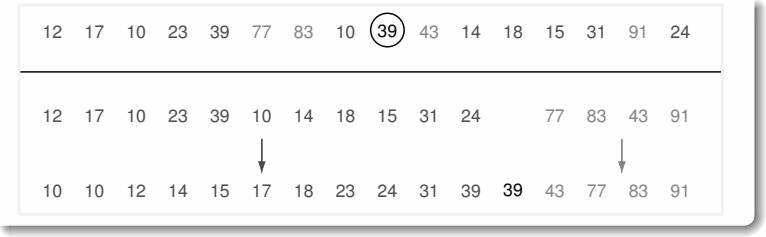
12 17 10 23 33 77 83 11 ⟶ ~~10~~ ~~11~~ ~~12~~ ~~17~~ ~~23~~ ~~33~~ ~~77~~ ~~83~~
39 43 14 18 15 31 91 24 ⟶ ~~14~~ ~~15~~ ~~18~~ ~~24~~ ~~31~~ ~~39~~ ~~43~~ ~~91~~

10 11 12 14 15 17 18 23 24 31 33 39 43 77 83 91

Coût du tri par dichotomie-fusion

$$\begin{cases} \mathrm{C}(1) = 1 \\ \mathrm{C}(n) = 2 \cdot \mathrm{C}(n/2) + O(n) \end{cases} \Rightarrow \mathrm{C}(n) = O(n \cdot \ln(n))$$

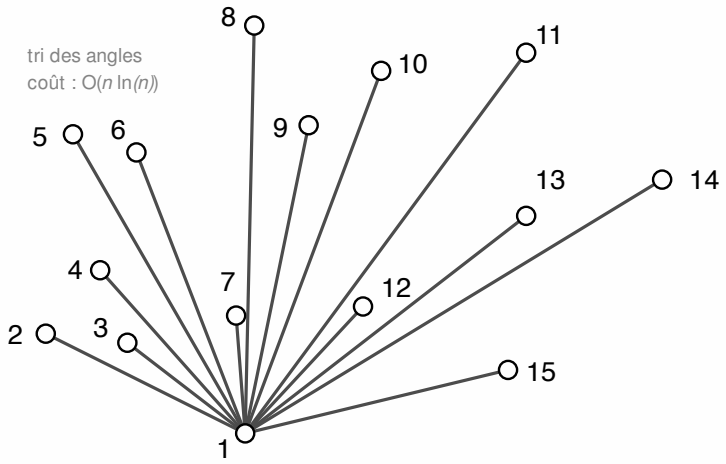
- choisir un élément du tableau (appelé **pivot**)
- le placer à sa place définitive en permutant tous les éléments afin que
  - tous ceux qui lui sont inférieurs soient à sa gauche
  - tous ceux qui lui sont supérieurs soient à sa droite
- pour chacun des sous-tableaux
  - définir un nouveau pivot
  - répéter l'opération de partitionnement
- cet algorithme récursif se termine quand les sous-tableaux sont vides



http://www.bluffton.edu/~nesterd/java/SortingDemo.html

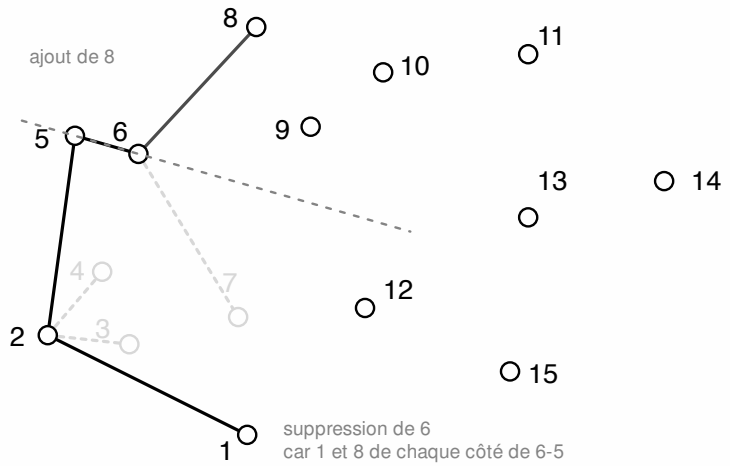
Propriété

Un segment est dans l'enveloppe convexe si et seulement si tous les autres points sont du même côté



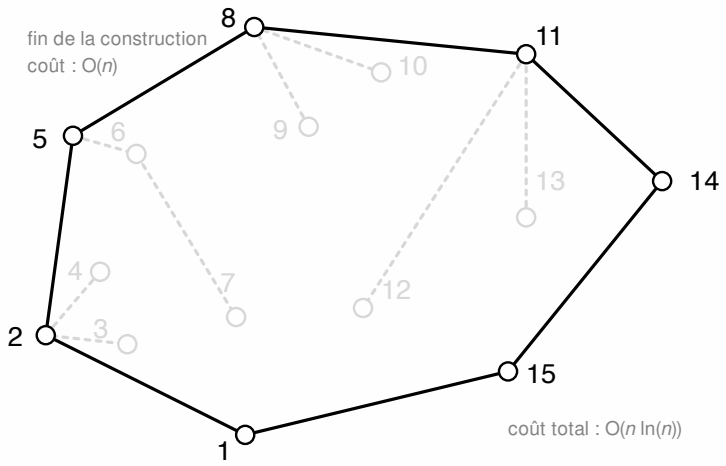
Propriété

Un segment est dans l'enveloppe convexe si et seulement si tous les autres points sont du même côté



Propriété

Un segment est dans l'enveloppe convexe si et seulement si tous les autres points sont du même côté



Plusieurs usages de l'aléatoire

- *algorithmes de Monte-Carlo* : peuvent calculer une solution incorrecte en un temps déterministe avec une probabilité d'erreur qui peut être rendue arbitrairement petite en répétant l'algorithme
- *algorithmes de Las Vegas* : ne terminent pas nécessairement, mais calculent toujours une solution correcte quand ils terminent, le temps d'exécution étant aléatoire

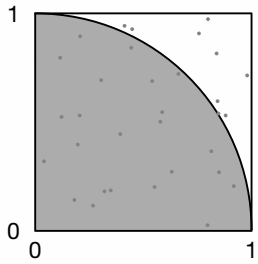
Approximation de la valeur de π

initialiser c à 0

répéter 10000 fois

- tirer une valeur aléatoire x dans $[0, 1]$
- tirer une valeur aléatoire y dans $[0, 1]$
- si $x^2 + y^2 < 1$, alors incrémenter c de 1

renvoyer $4 \times \dfrac{c}{10000} = \left(4 \times \dfrac{\mathcal{A}(\text{¼ disque})}{\mathcal{A}(\text{carré})}\right)$



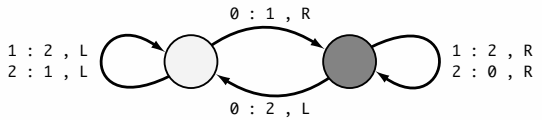
- machine abstraite inventée en 1936 par Alan Turing (1912-1954), pour servir de modèle idéal lors d'un calcul mathématique
- tous les ordinateurs modernes sont conçus selon son principe de fonctionnement

Propriétés d'un *modèle formel* permettant de décrire un algorithme

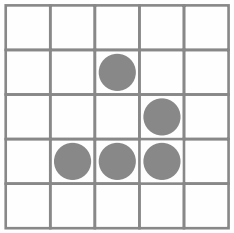
- chaque procédure doit recevoir une définition finie
- la procédure doit être composée d'étapes distinctes, dont chacune doit pouvoir être accomplie mécaniquement

Composition d'une machine de Turing abstraite

- une *mémoire infinie* représentée par un ruban divisé en cases, chacune pouvant recevoir un symbole de l'alphabet défini pour la machine
- une *tête de lecture/écriture* capable de parcourir le ruban dans les deux sens
- un *ensemble fini d'états* (dont un état initial et des états accepteurs)
- une *fonction de transition* qui, pour chaque état de la machine et chaque symbole figurant sous la tête de lecture, précise
  - l'état suivant
  - le caractère à écrire sur le ruban (en remplacement de celui lu)
  - le sens du prochain déplacement de la tête



http://interstices.info/jcms/c_43049/machine-de-turing



- automate cellulaire imaginé en 1970 par John Horton Conway (1937-)
- modèle composé d'une grille dont les *cellules*
  - sont à chaque instant soit vivantes, soit mortes
  - évoluent selon des règles pré-établies très simples

# voisins	0	1	2	3	4	5	6	7	8
naissance				✔					
survie			✔	✔					

Exemples d'application

- simulation du comportement d'un gaz
- étude des matériaux magnétiques selon le modèle d'Ising
- simulation des processus de percolation, de cristallisation
- alternative aux équations différentielles
- conception d'ordinateurs massivement parallèles
- simulation et étude du développement urbain
- simulation de la propagation des feux de forêt

http://golly.sourceforge.net/