

Information, Calcul et Communication

Introduction

Olivier Lévêque et Mirjana Stojilovic

Pourquoi un cours d'introduction à l'informatique ?

- **4e pilier** de la culture (après la lecture, l'écriture et l'arithmétique)
- Elle constitue désormais une **discipline scientifique à part entière**: la science du traitement automatique de l'information.
- L'informatique a non seulement changé notre société, mais aussi **notre façon de faire de la science**.
- De nos jours, tout.e ingénieur.e qui maîtrise les sciences du numérique a clairement un avantage sur les autres...

Plan du cours (partie théorique)

Première partie : Calcul (introduction aux algorithmes)

- Ingrédients de base
- Complexité temporelle
- Récursivité
- Programmation dynamique
- Calculabilité
- Classes de complexité
- Méthodes d'approximation

Seconde partie : Information et communication

- Représentation de l'information
- Architecture des ordinateurs
- Échantillonnage et reconstruction de signaux
- Entropie et compression de données
- Correction d'erreurs
- Réseaux
- Cryptographie et sécurité

Horaires (partie théorique)

- **Cours :**

les vendredis après-midis de 14h15 à 16h
en salles SG 1, SG 0123 (retransmission) et sur Zoom

- **Exercices:**

les vendredis après-midis de 16h15 à **17h15+** en salles DIA 004 / DIA 005 /
INF 1 / INF 119 / INJ 218 / INM 201 / INM 202 / INM 203 (!)

Une vingtaine de personnes sont là pour vous : ***profitez-en !***

- Pour la partie théorique:

un examen final seulement, valant pour 50% de la finale

- Pour la partie programmation:

un midterm, un devoir noté et un examen final,
valant pour 50% de la note finale

- **EPFL !**
- **Moodle** : matériel de cours, vidéos, exercices, corrigés, références, ...
(de manière générale, vous trouverez là *toutes* les informations sur le cours)
- **Chaîne Mediaspace** avec vidéos pré-enregistrées du cours
- **Zoom** : cours retransmis en direct, enregistrements du cours
- **Forum EdDiscussion** : vous pouvez poser des questions à tout moment, de manière anonyme si vous le désirez; encore une fois, ***profitez-en !***

EPFL Références

- Livre ``Découvrir le numérique'', EPFL Press, 2016
- Fichiers pdf avec transparents du cours, plus quelques compléments

Encore quelques conseils...

(que vous connaissez déjà sans doute)

- Votre participation active au cours et aux exercices est cruciale !
- Prenez des notes !
- N'hésitez pas à poser des questions ! pendant le cours aussi !
- Retravaillez le cours et les exercices après les séances...

Qu'est-ce qu'un algorithme ?

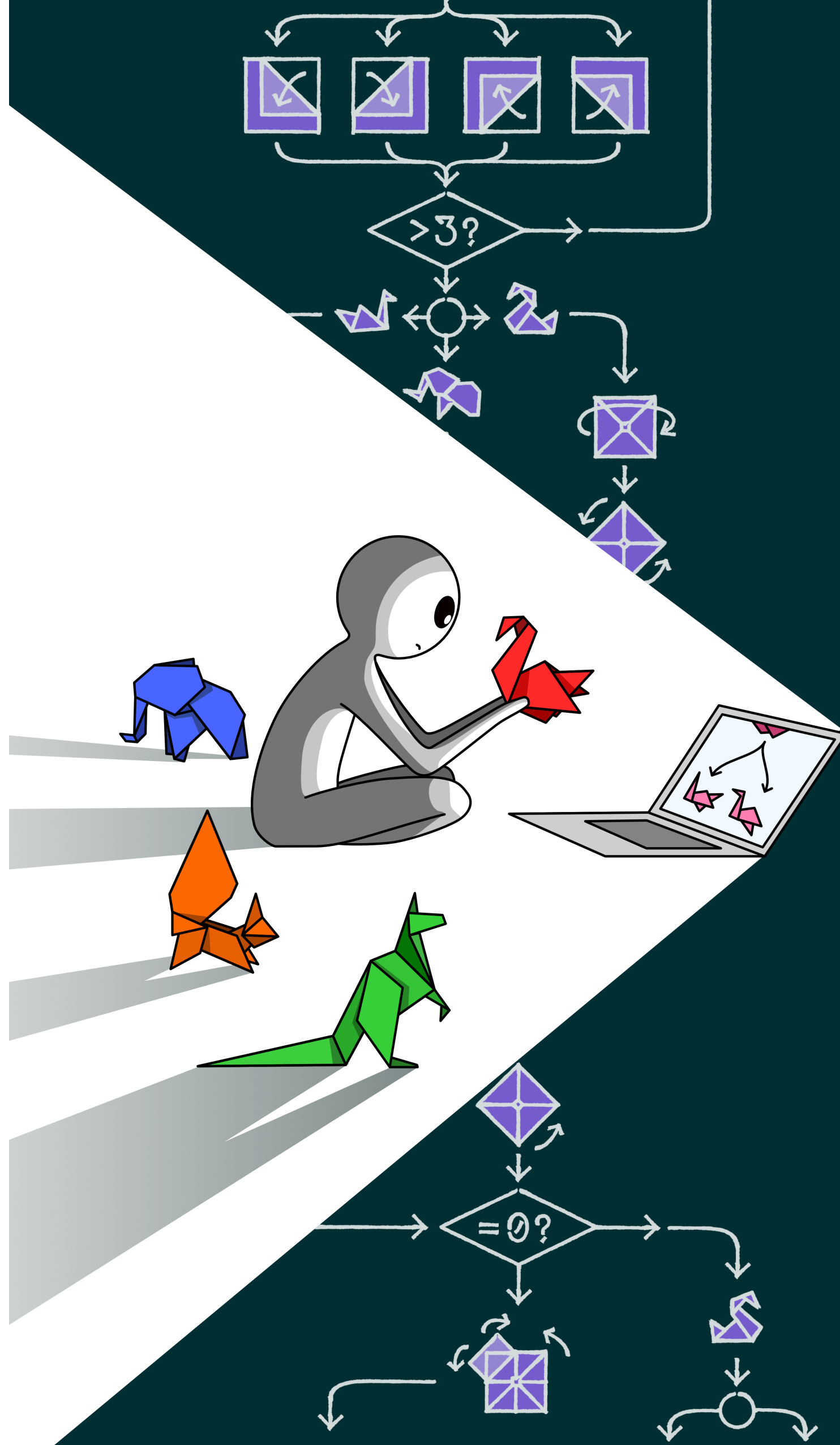
- Un algorithme n'est **pas** un programme.
- Un algorithme est la description des étapes **élémentaires** menant à la résolution d'un problème; c'est donc la description conceptuelle d'un programme.
- Un **programme** est l'implémentation d'un algorithme dans un langage donné et dans un système particulier.

EPFL Exemple 1: calcul du modulo 3 d'un grand nombre

EPFL Exemple 2: recherche du minimum dans une liste

Exemple 3: problème du voyageur de commerce





Information, Calcul et Communication

Algorithmes :
ingrédients de base

Olivier Lévêque

EPFL Algorithmes : ingrédients de base

Données

- Entrées
- Sorties
- Variables internes

Instructions

- Affectations
- Structures de contrôle
 - Branchements conditionnels (tests)
 - Itérations (boucles)
 - Boucles conditionnelles

- **Question :**

Est-ce que tous les objets visibles sur cette photo sont différents les uns des autres ?

- **Question réciproque :**

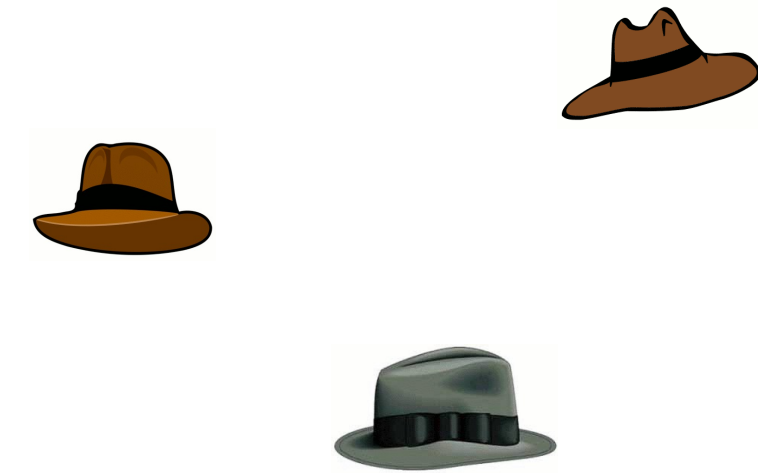
Y a-t-il au moins deux objets identiques sur cette photo ?



Tous différents ?

Problème à résoudre:

Parmi une liste de 3 objets, identifier si ceux-ci sont tous différents les uns des autres.



Algorithme

entrée : $L = (L(1), L(2), L(3))$ liste de 3 objets
sortie : valeur binaire oui/non

```
 $s \leftarrow \text{oui}$   
Si  $L(1) = L(2)$ , alors :  $s \leftarrow \text{non}$   
Si  $L(1) = L(3)$ , alors :  $s \leftarrow \text{non}$   
Si  $L(2) = L(3)$ , alors :  $s \leftarrow \text{non}$   
Sortir :  $s$ 
```


Algorithme d'Euclide

L'algorithme d'Euclide utilise une boucle conditionnelle pour trouver le plus grand diviseur commun (pgdc) de deux nombres entiers.

Algorithme

entrée : a, b deux nombres entiers positifs

sortie : $pgdc(a, b)$

Tant que $b \neq 0$:

$temp \leftarrow b$

$b \leftarrow a \bmod b$

$a \leftarrow temp$

Sortir : a