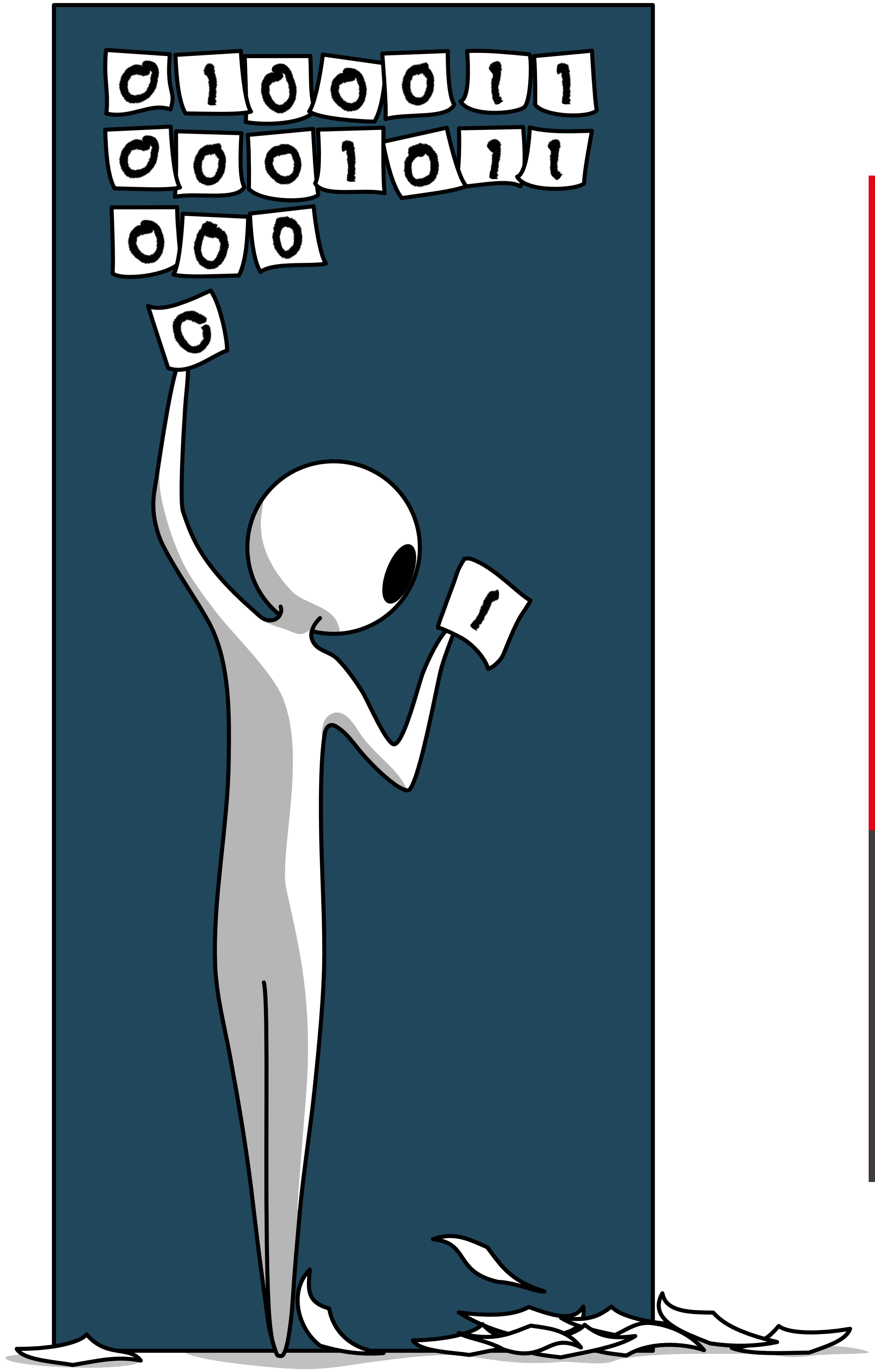
EPFL



Information, Calcul et Communication

Introduction

Olivier Lévêque
et
Dan-Cristian Tomozei

École
polytechnique
fédérale
de Lausanne

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
- Echantillonnage
  et reconstruction de signaux
- Entropie
  et compression de données
- Correction d'erreurs
- Cryptographie et sécurité

Horaires (partie théorique)

- **Cours :**

  les vendredis après-midis de 13h15 à 15h
  en salle SG 1 et sur Zoom

- **Exercices:**

  les vendredis après-midis de 15h15 à **16h15+** en salles:

  Lettres A à H: CO 010, CO 011, CO 015, CO 016, CO 017
  Lettres J à Z: CO 120, CO 121, CO 122, CO 123, CO 124

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

Evaluation

- Mini-projet de programmation, valant pour 15% de la note finale
- Quiz sur la sécurité informatique, valant pour 5% de la note finale
- Examen final en juin-juillet, portant sur les deux parties du cours (théorie et programmation), valant pour 80% de la note finale.

Points de contact

- **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 !***

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.

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

1

$$
\begin{array}{r|l}
47'256 & 3 \\ \hline
-3 & 15752 \\
17 & \\
-15 & \\
22 & \\
-21 & \\
15 & \\
-15 & \\
06 & \\
-6 & \\
0 \leftarrow \text{modulo } 3 & \\
\in\{0,1,2\} &
\end{array}
$$

2

$$\underline{4+7+2+5+\cancel{6}} = \underline{\underline{24}}$$

$$
\begin{array}{r|l}
24 & 3 \\ \hline
-24 & 8 \\
0 &
\end{array}
$$

expl:

$$47'256 = 4\cdot 10'000 + 7\cdot 1'000 + 2\cdot 100$$

$$+ 5\cdot 10 + 6\cdot 1$$

$$10'000 = \underline{9'999} + 1$$

2bis

$$2+4=6$$

$$- - - -$$

Exemple 2: recherche du minimum dans une liste

longueur n

donnée d'entrée: $L = (17, 18, 24, 3, 15, 12, 9, 8, 7, 4, 2, 18, 24)$

1. on compare le 1er & le 2e nombre ; on garde le plus petit
2. on compare 17 avec le suivant ; on garde le plus petit
3. on continue ...

$a \leftarrow L(1)$
$i \leftarrow 2$
Tant que $i \leq n$:
  si $a > L(i)$, alors $a \leftarrow L(i)$
  $i \leftarrow i+1$
Sortir a $\longrightarrow$ a = minimum de la liste L

données d'entrée: $\begin{cases} \text{liste } L \\ \text{taille } n \end{cases}$

donnée de sortie: minimum

Note : algorithme de recherche du minimum d'une liste :

entrée : liste L, taille n

sortie : minimum de la liste (a)

V1 :

$a \leftarrow L(1)$

$i \leftarrow 2$

Tant que $i \leq n$:

  si $L(i) < a$, alors

    $a \leftarrow L(i)$

  $i \leftarrow i+1$

Sortir a

V2 :

$a \leftarrow L(1)$

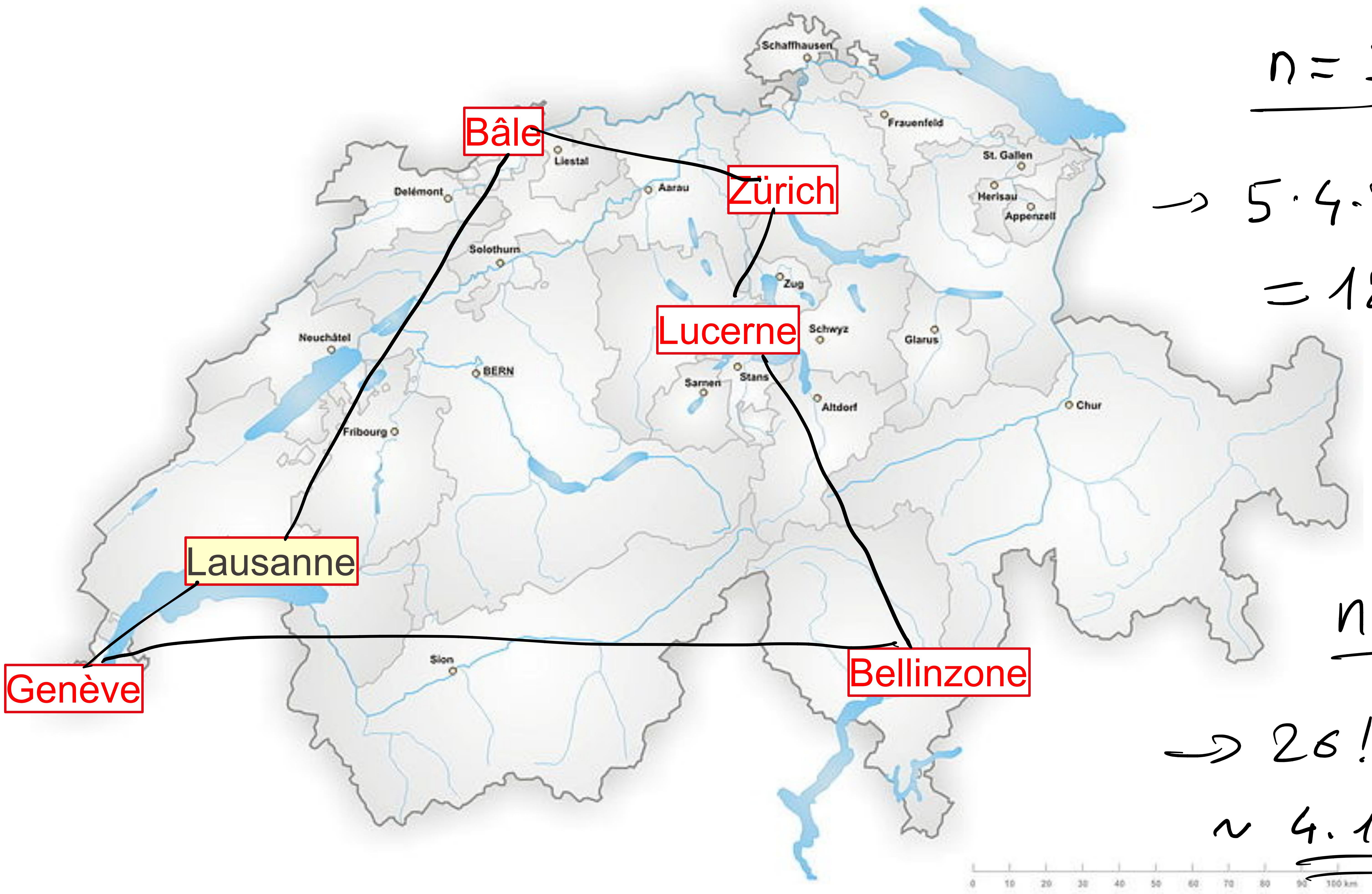
Pour i allant de 2 à n :

  si $L(i) < a$, alors

    $a \leftarrow L(i)$

Sortir a

Exemple 3: problème du voyageur de commerce



$n = 5$

$\rightarrow 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5!$

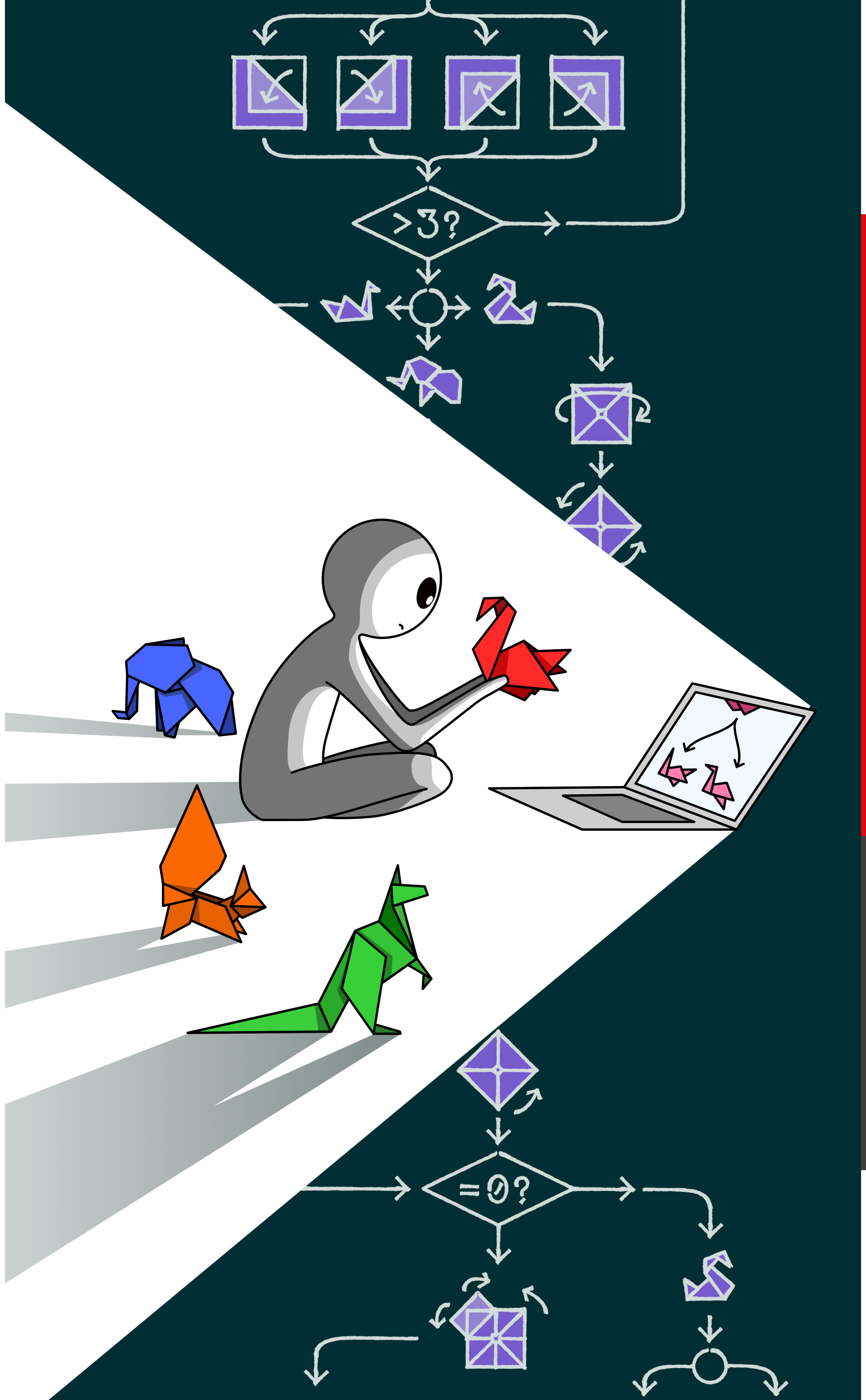
$= 120$ possibilités

$n = 26$

$\rightarrow 26!$ possibilités

$\sim 4 \cdot 10^{26}$

EPFL



Information, Calcul et Communication

Algorithmes : ingrédients de base

Olivier Lévêque

École polytechnique fédérale de Lausanne

EPFL

Algorithmes : ingrédients de base

Données

- Entrées ← a, b, c
- Sorties ← $\{x \in \mathbb{R} : ax^2+bx+c=0\}$
- Variables internes ← Δ

Instructions

- Affectations ($\Delta \leftarrow b^2-4ac$)
- Structures de contrôle
  - Branchements conditionnels (tests) si ⌴ (condition), alors ___ sinon ___
  - Itérations (boucles) → Pour i allant de 1 à n, ... (répéter)
  - Boucles conditionnelles

↳ Tant que ⌴ (condition), ... (répéter)

Exemple : $a \neq 0$

Pb : étant donné $a, b, c \in \mathbb{R}$ trouver $\{x \in \mathbb{R} : ax^2+bx+c=0\}$

$\Delta \leftarrow b^2-4ac$

si $\Delta<0$, sortir $\emptyset$

sinon, si $\Delta=0$, sortir $\{-\frac{b}{2a}\}$

sinon, sortir $\{\frac{-b\pm\sqrt{\Delta}}{2a}\}$

Illustration

- **Question :**

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

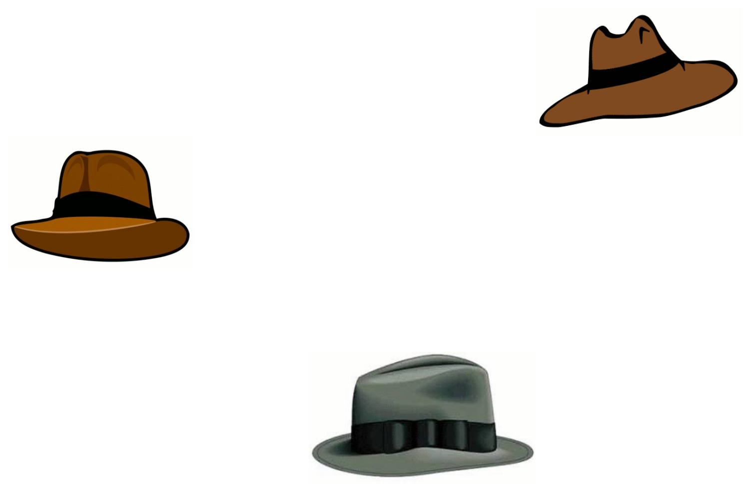
- **Question réciproque :**

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

Si $L(1) = L(2)$, sortir non
Sinon, si $L(1) = L(3)$, sortir non
Sinon, si $L(2) = L(3)$, sortir non
Sinon, sortir oui

si $L(1) = L(2)$, sortir non
si $L(1) = L(3)$, sortir non
si $L(2) = L(3)$, sortir non
Sortir oui

~~Si $L(1) \neq L(2)$, sortir oui~~

~~Si $L(1) \neq L(3)$, sortir oui~~

~~Si $L(2) \neq L(3)$, sortir oui~~

~~Sortir non~~

Si $L(1) \neq L(2)$ & $L(1) \neq L(3)$ & $L(2) \neq L(3)$,

sortir oui

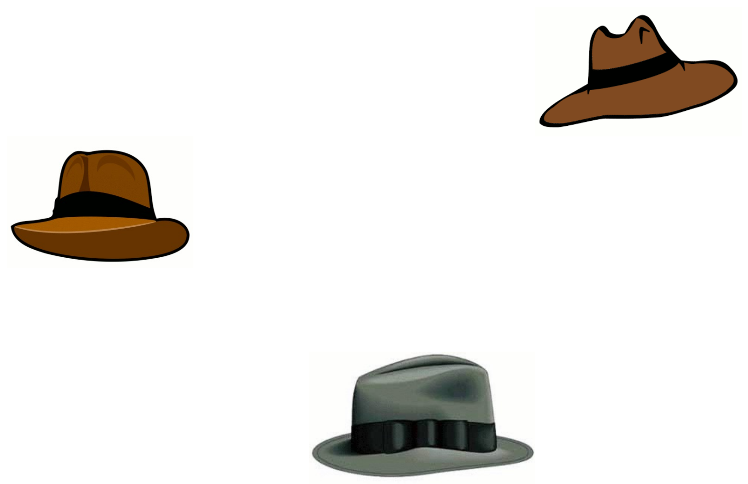
✓

Sinon, sortir non

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

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

Tous différents ? (bis)

Problème à résoudre:

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

Algorithme

entrée : L liste de n objets, n taille de la liste
sortie : valeur binaire oui/non

Pour i allant de 1 à $n-1$
    Pour j allant de $\underline{i+1}$ à n
        Si $L(i) = L(j)$, sortir non
Sortir oui

$i=1$: $j=2$ --- $j=n$
$i=2$: $j=3$ --- $j=n$

Tous différents ? (bis)

Problème à résoudre:

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

Algorithme

entrée : L liste de n objets, n taille de la liste
sortie : valeur binaire oui/non

$s \leftarrow$ oui
Pour i allant de 1 à $n - 1$:
    Pour j allant de $i + 1$ à n :
        Si $L(i) = L(j)$, alors : $s \leftarrow$ 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

$pgdc(a, b)$
$= pgdc(b, a \bmod b)$

$a = 30, \; b = 12 \implies$ ---