

UE 4I900  
Master Informatique, UPMC

# COMPLEX

COMPLEXité, algorithmes randomisés et  
approchés

Enseignants : B. Escoffier, F. Pascual,  
L. Perret (responsable),

# Contenu de l'UE

## **S 1-5 :**

- \* Introduction à la théorie du calcul et à la complexité des problèmes.
- \* Méthodes arborescentes.
- \* Algorithmes d'approximation.

## **S 6-10 :**

- \* Algorithmes probabilistes, classes de complexité probabilistes.  
En particulier : test de primalité

# Organisation et contrôle des connaissances

- Un cours (2h) par semaine
- 2h TD-2h TME, ou 4h TD par semaine
- Un projet pendant les 5 derniers TME :
  - En binômes, langage au choix
  - Soutenances les semaines 9 et 10 de TME.
- Site de l'UE :  
<http://www-poleia.lip6.fr/~pascualf/complex2016>

# Organisation et contrôle des connaissances

- Projet (rapport+soutenance) : 20 %
- Examen réparti 1 (novembre) : 40 %
- Examen réparti 2 (janvier) : 40 %

# Chapitre 1

## Introduction à la théorie du calcul

a. Y a-t-il des **problèmes** que l'on ne **peut pas** résoudre avec un ordinateur, i.e. pour lesquels il n'existe pas d'**algorithme** ?

*Problème ?*

*Algorithme ?*

# Introduction à la théorie du calcul

**b.** Parmi les problèmes que l'on peut résoudre, y en a-t-il de plus difficiles que d'autres ?

GPS



Trajet optimal en quelques secondes

Eternity II



Puzzle  $16 \times 16$

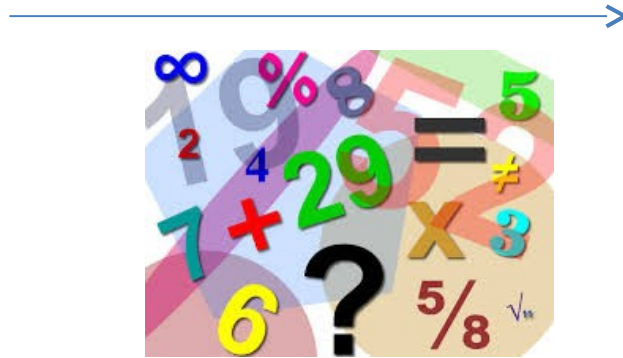
2 000 000 \$

# Introduction à la théorie du calcul

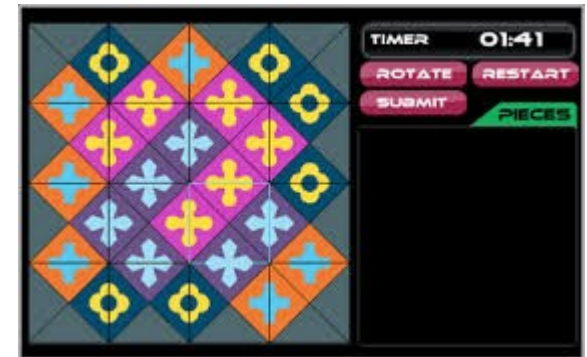
## I. Problème, algorithme



- Une donnée/  
une instance
- Une question



Une séquence  
d'instructions/  
un algorithme



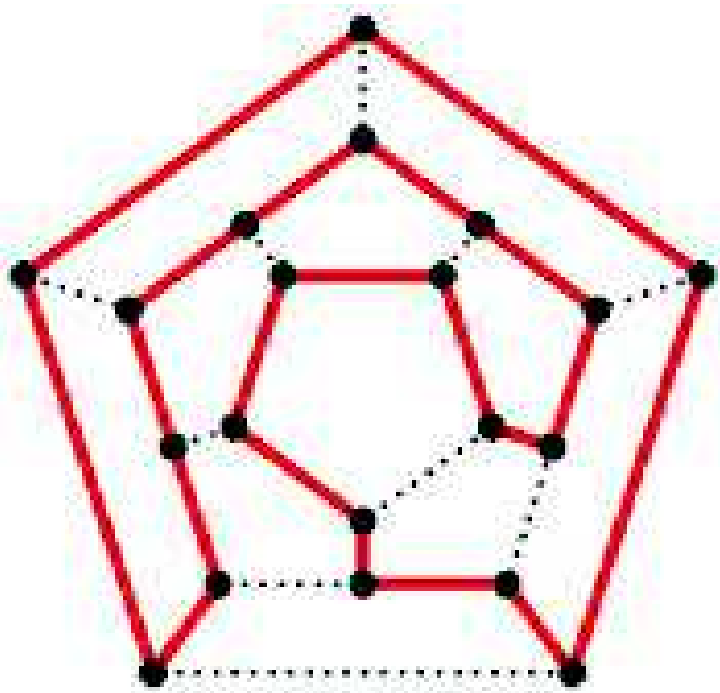
Une réponse/  
une solution

## Exemple 1 : chaîne entre deux sommets

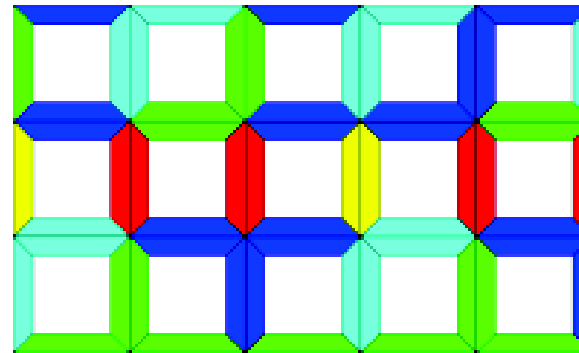
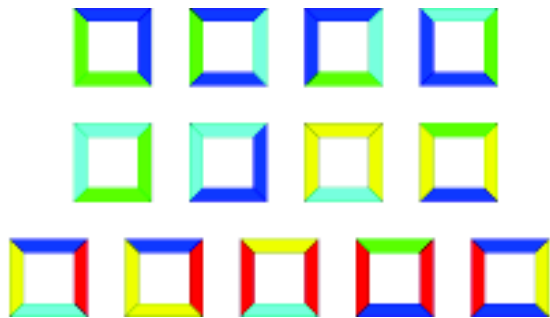




## Exemple 2 : chaîne hamiltonienne



## Exemple 3 : pavage du plan



...

...

# I. Problème, algorithme



Muhammad ibn Mūsā Al-Khwārizmī  
(environ 780-850, Ouzbékistan)

Algorithme : Suite finie et non ambiguë d'instructions simples.

But : donner la réponse à une question (résoudre un problème).

# I. Problème, algorithme



David Hilbert  
(1862-1943, Prusse/Allemagne)

Congrès international des  
mathématiciens, Paris, 1900.  
23 problèmes ouverts.

**Enoncé 10 :** Trouver un  
algorithme déterminant si  
un polynôme à coefficients  
entiers a une racine entière.

## II. Formalisation



Alan Turing  
(1912-1954, Angleterre)

- 1936 : la machine de Turing  
Article fondateur de la science informatique : « *On Computable Numbers, with an Application to the Entscheidungsproblem* ».
- 1936 : décodage du code secret allemand ENIGMA.
- 1950 : intelligence artificielle.

## II. Formalisation

### 1. Problème et langage

## II. Formalisation

1. Problème et langage
2. Machine de Turing

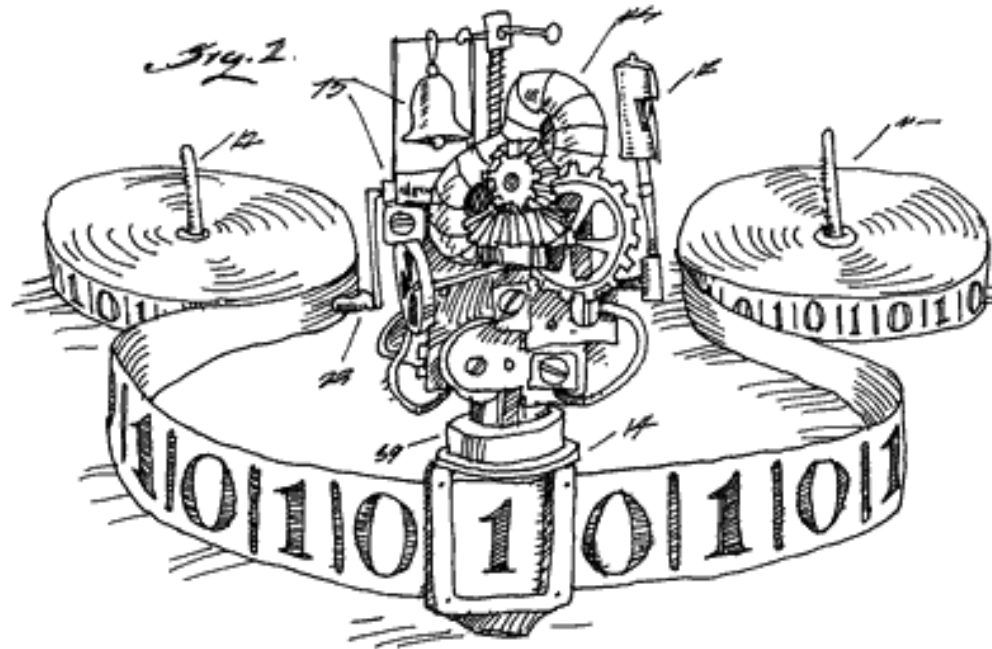
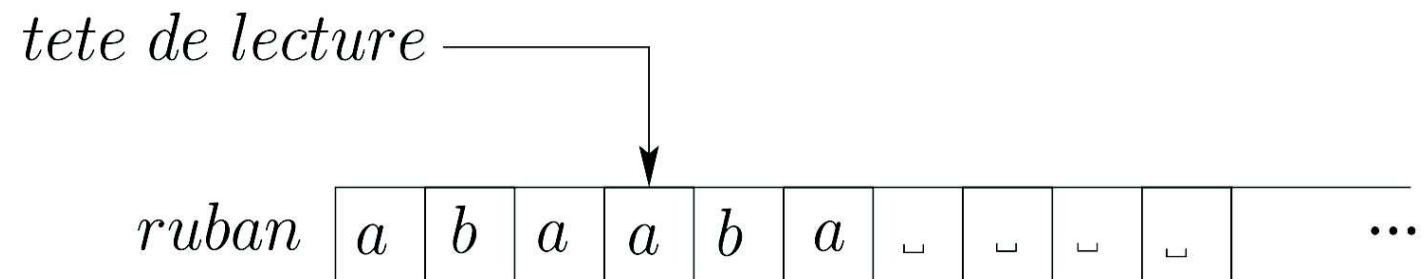


Schéma d'une machine de Turing :

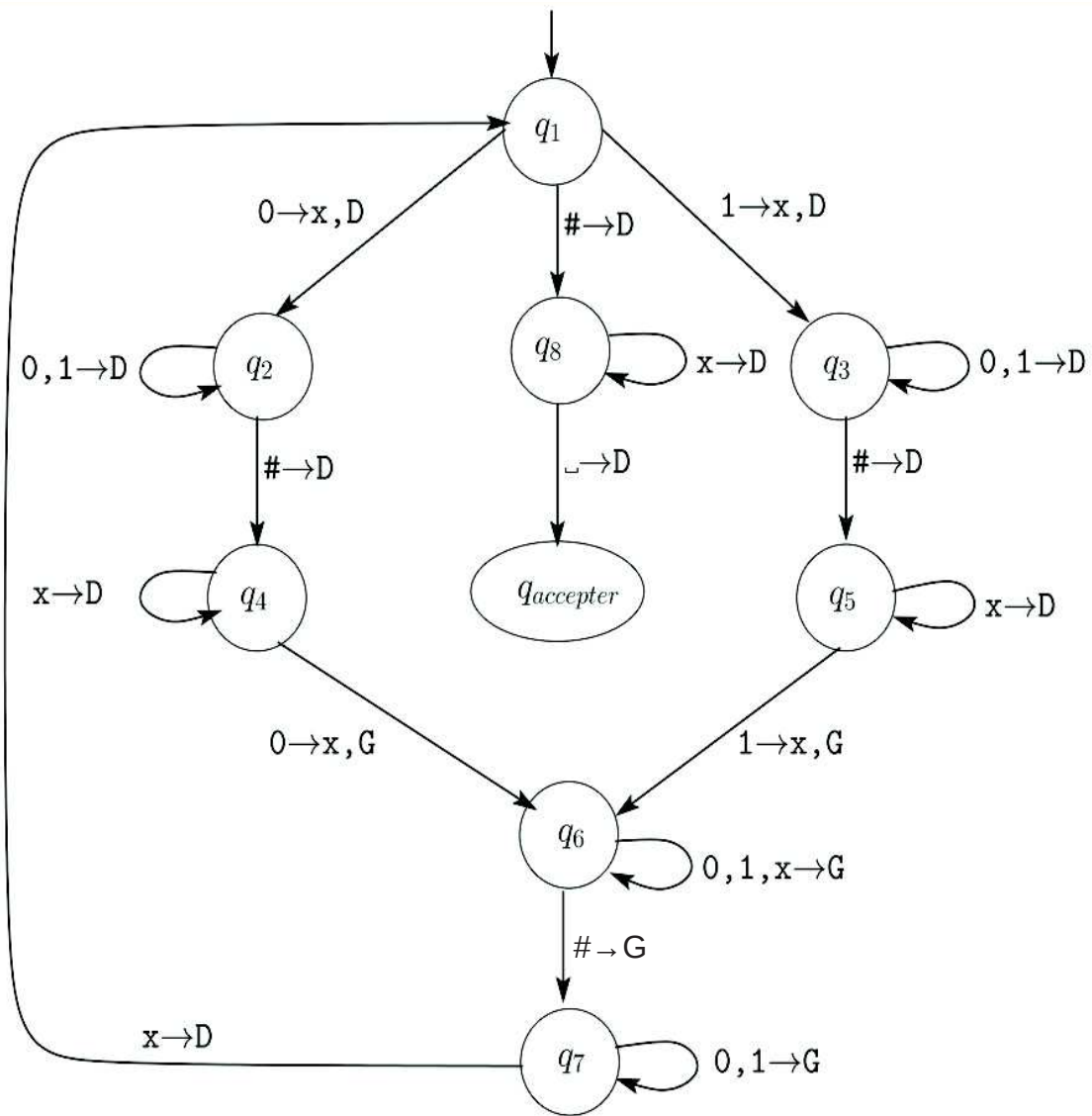




## Exemple 1 :

Problème/langage :  $\{w#w : w \text{ mot sur } \{0,1\}\}$

[illegible]

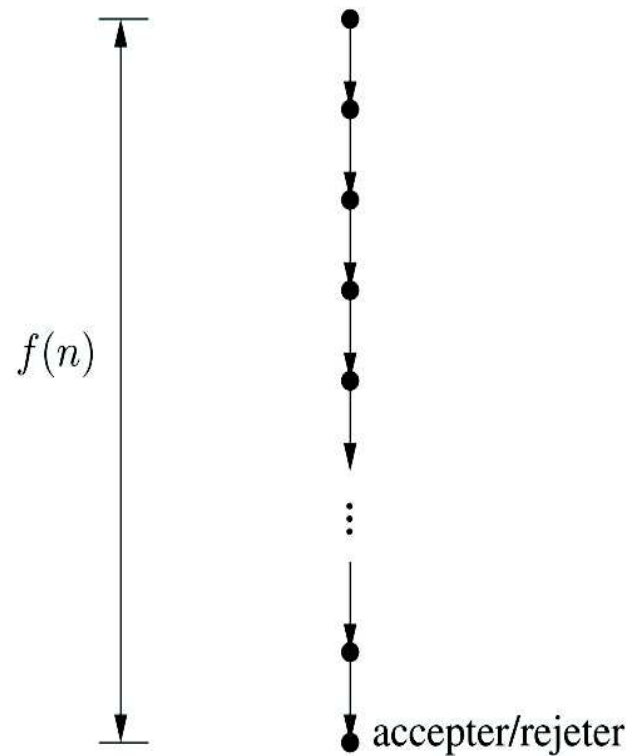


# Problème/langage : éléments distincts

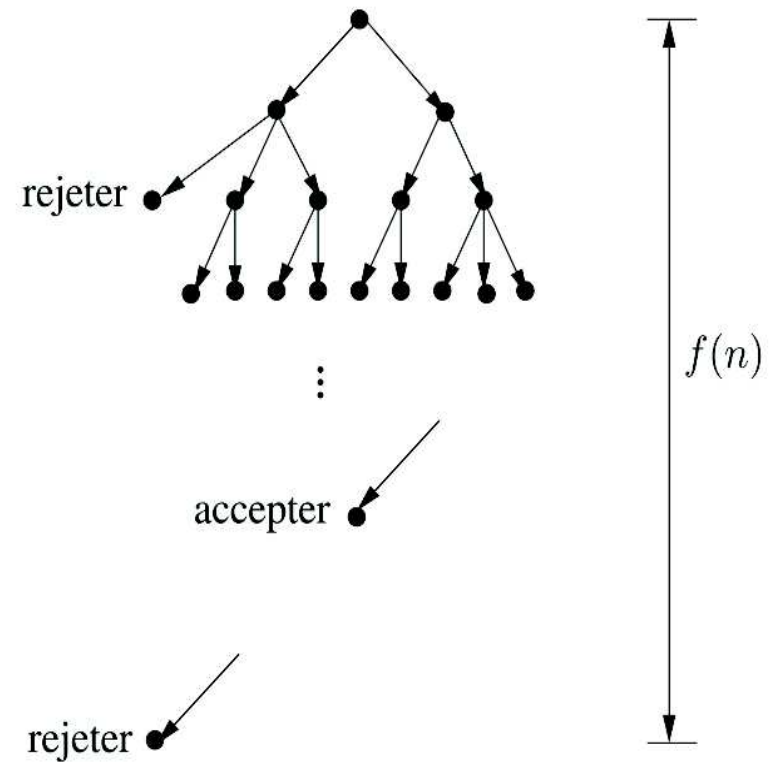
•  
•  
•

### 3. Machine de Turing non déterministe

MT deterministe



MT non deterministe



# Remarques pour conclure

- D'autres modèles ?

Thèse de Church (Turing)

