

UE 4I900

Master Informatique, Sorbonne Université

COMPLEX

COMPLEXité, algorithmes randomisés
et approchés

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.

Organisation

- Un cours (2h) par semaine.
- 4h TD par semaine les 2 premières semaines, 2h TD et 2h TME les semaines 3 à 10.
- Un projet pendant les TMEs de la deuxième partie de l'UE.
- Ressources sur moodle.

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 16×16

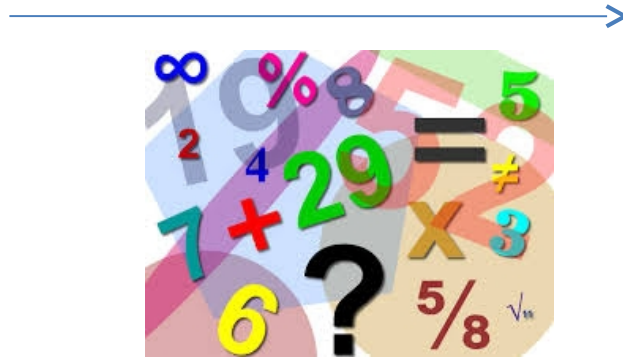
2 000 000 \$

Introduction à la théorie du calcul

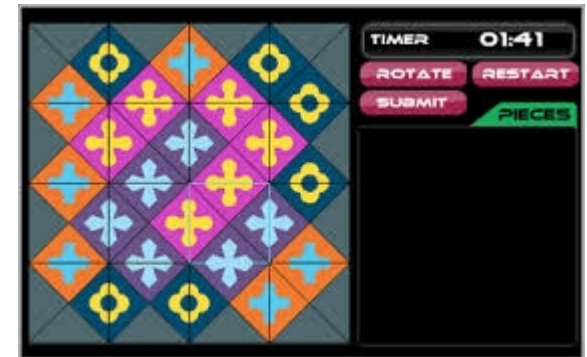
I. Problème, algorithme



- Une donnée/
une instance
- Une question



Une séquence
d'instructions/
un algorithme

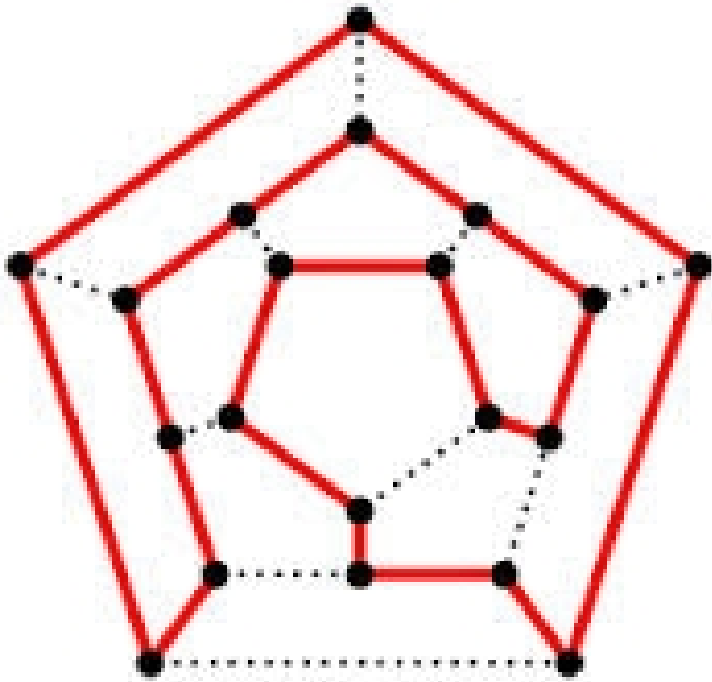


Une réponse/
une solution

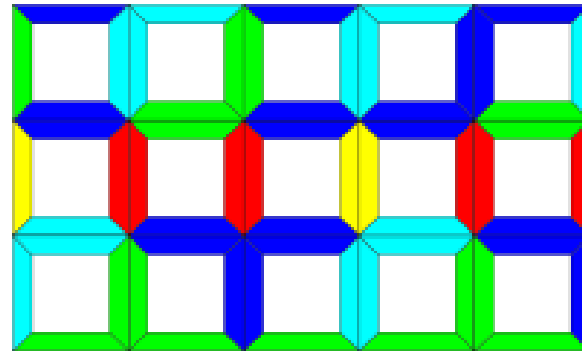
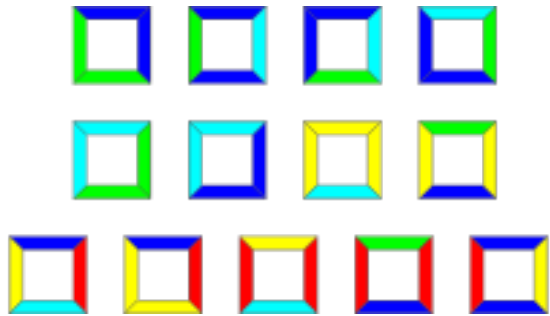
Exemple 1 : chaîne entre deux sommets



Exemple 2 : chaîne hamiltonienne



Exemple 3 : pavage du plan



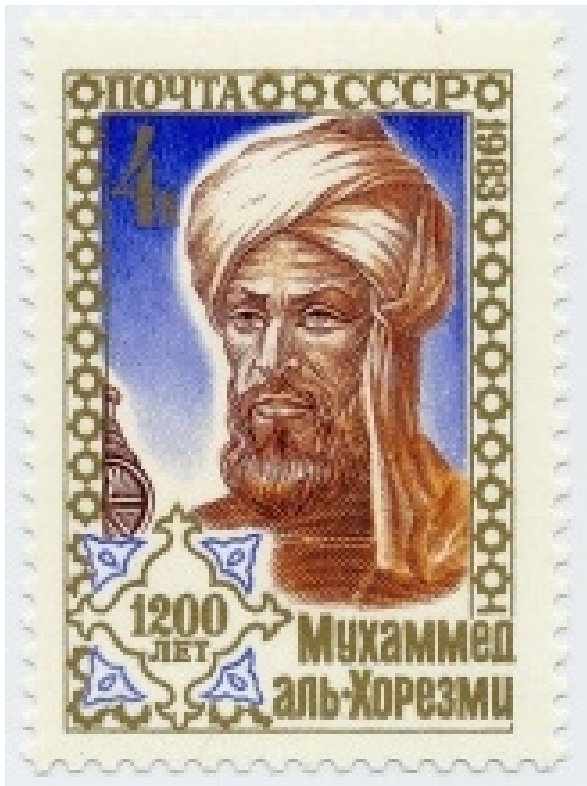
...

...

I. Problème, algorithme

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

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



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

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

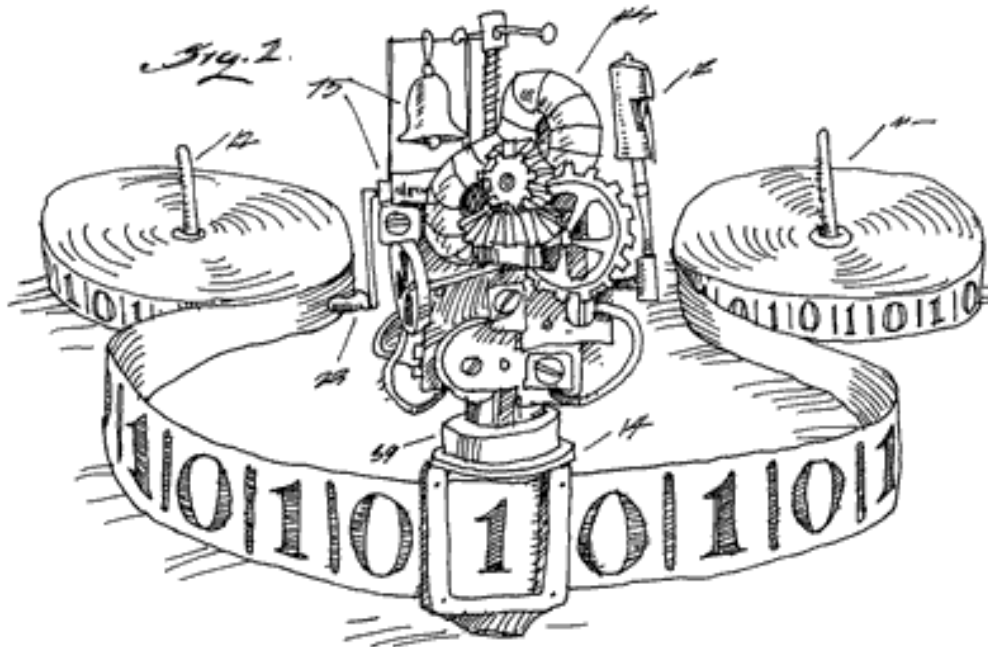
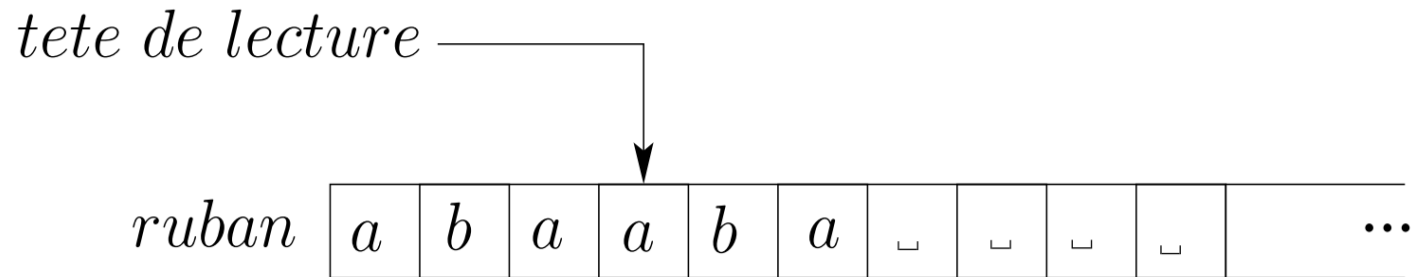
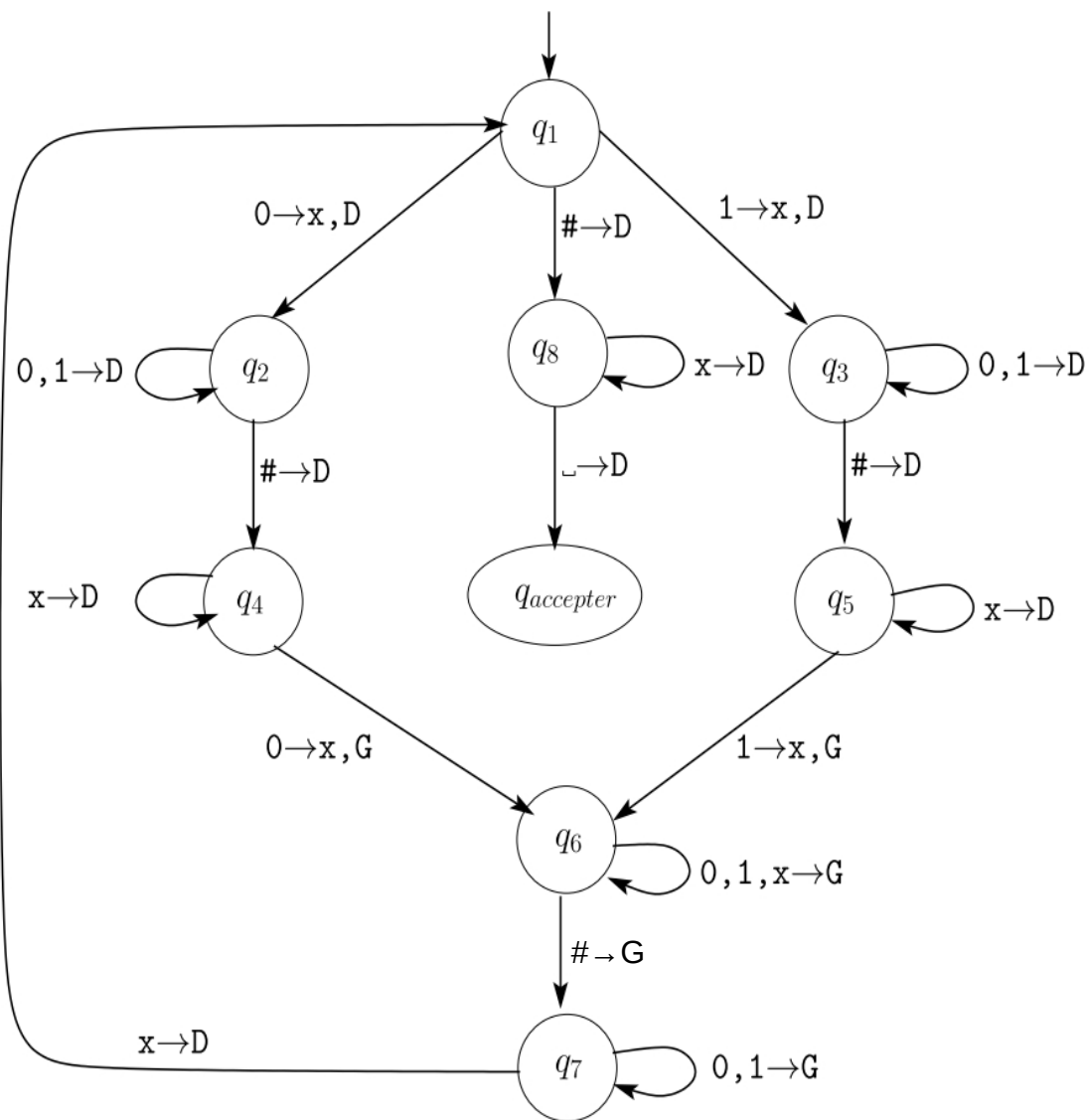


Schéma d'une machine de Turing :



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

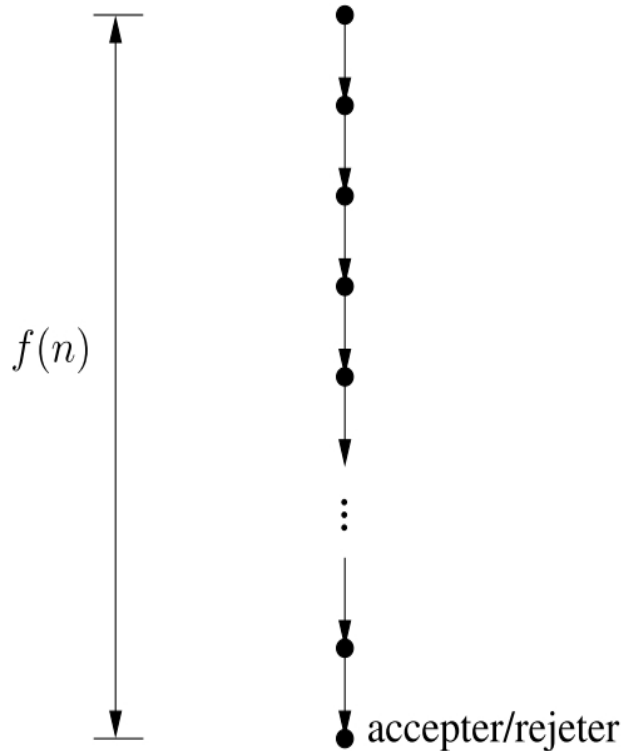


Problème/langage : éléments distincts

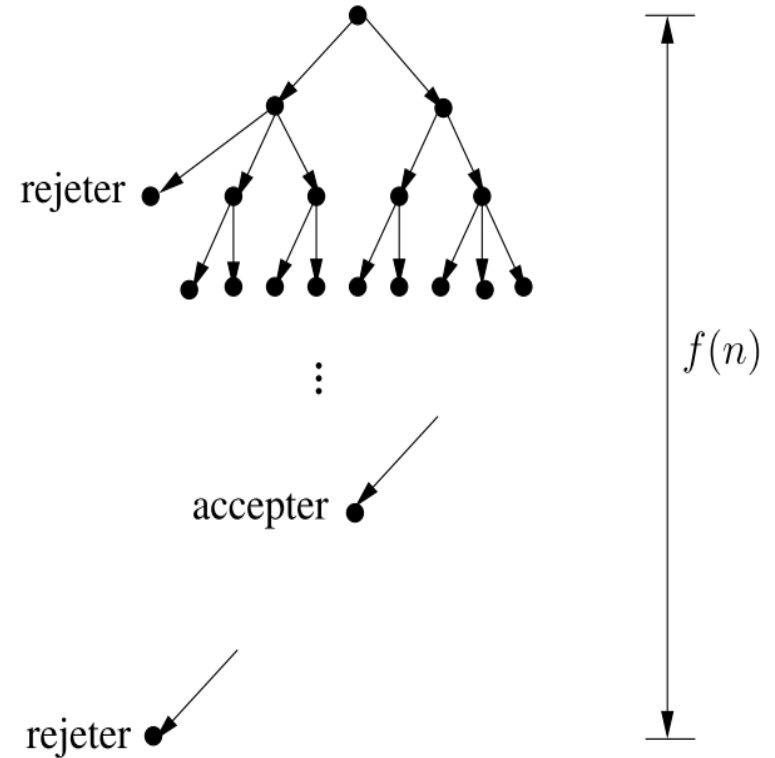
⋮

3. Machine de Turing non déterministe

MT déterministe



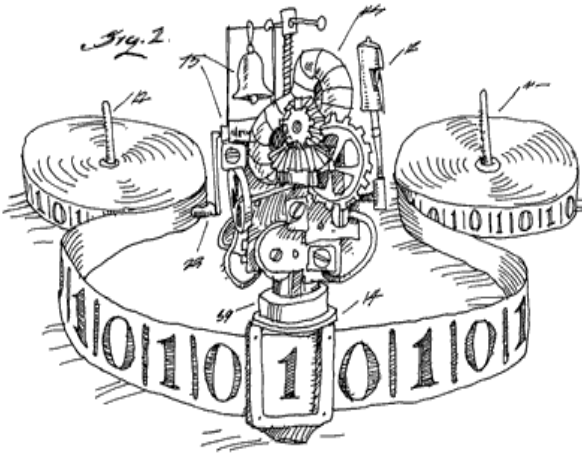
MT non déterministe



Remarques pour conclure

D'autres modèles ?

Thèse de Church (Turing)



=

