The P vs. NP problem

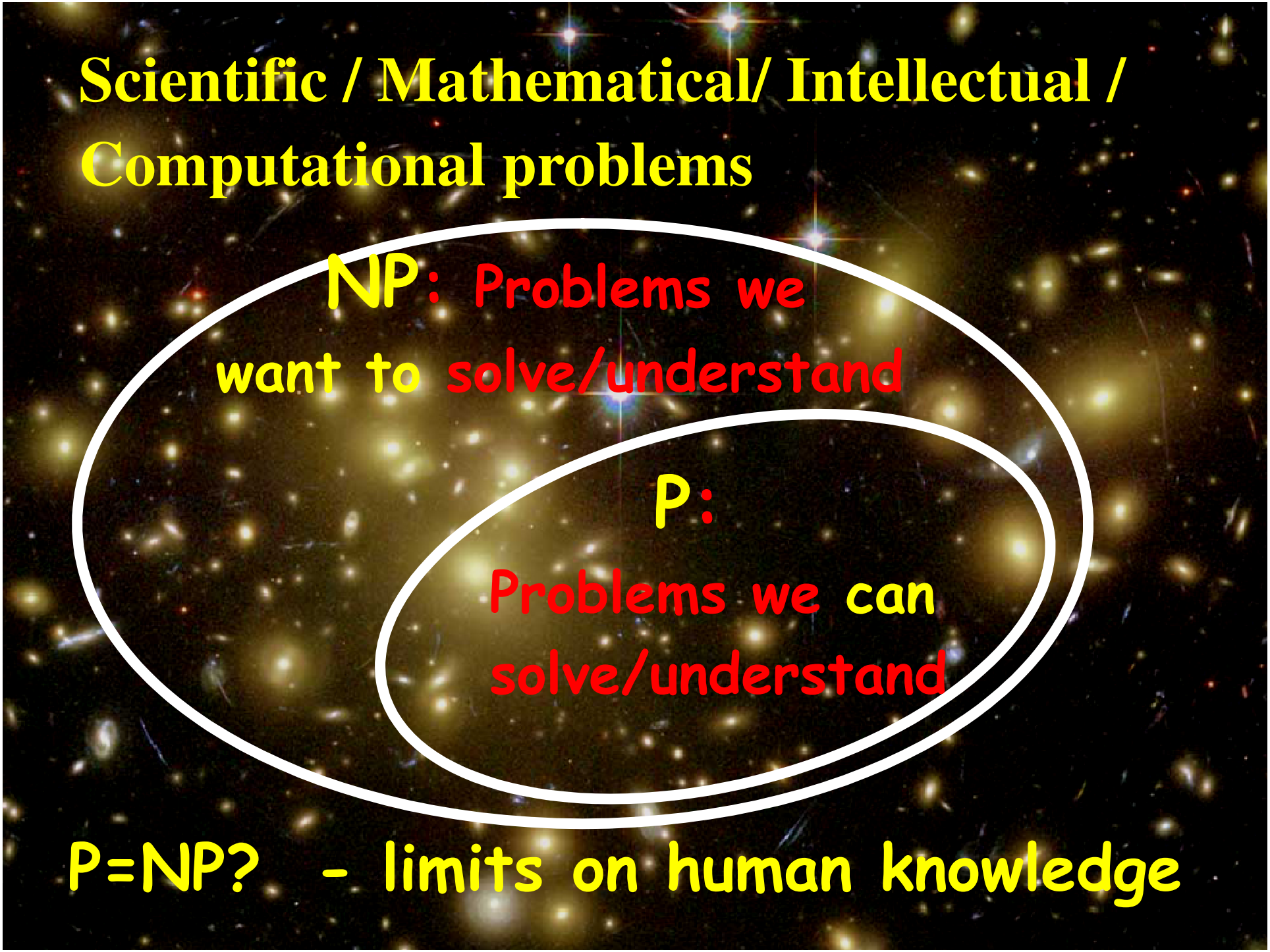
Efficient computation, Internet security, and the limits of human knowledge

Avi Wigderson
Institute for Advanced Study

Clay Math Institute Millennium Problems - $1M each

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- P vs. NP

- ~~Poincaré Conjecture~~
- Riemann Hypothesis
- Yang-Mills Theory


Scientific / Mathematical/ Intellectual / Computational problems
NP: Problems we want to solve/understand
P: Problems we can solve/understand
P=NP? - limits on human knowledge

PLAN

- Computation is everywhere
- Algorithms: language of computation
- Efficient algorithms: P
- Efficient verification: NP
- NP-completeness
- Implications

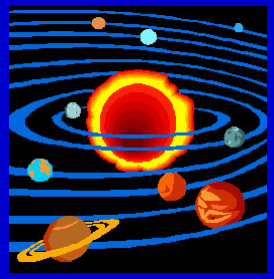
Computation

Mathematics

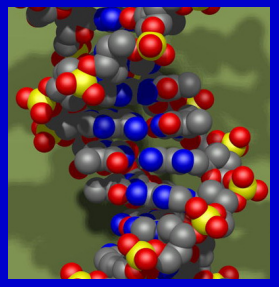
$$X^n + Y^n = Z^n$$

Computer Science

Computation



Physics



Biology

Computation is everywhere

Computation: every **process** which is a sequence of *simple, local* steps, that we want to **perform**, or **understand**

Variety of natural phenomena and intellectual challenges, each with an essential **computational** component

1 month

input

2 pm

Fetal development

Weather evolution

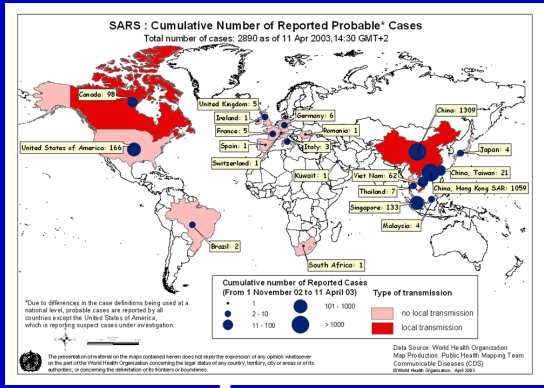
3 months

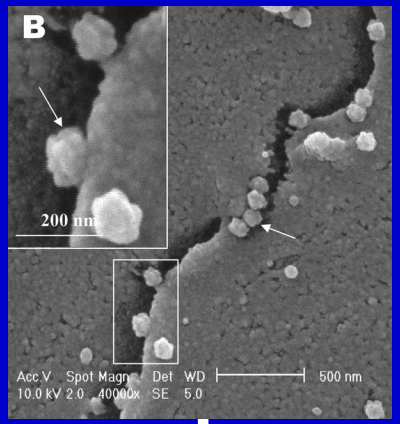
output

4 pm

Nature computes !
Can we simulate/predict?

4/11/03



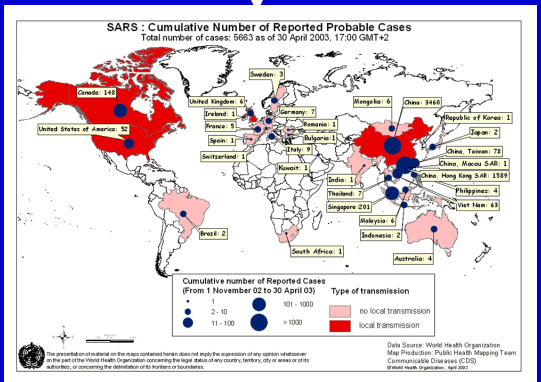


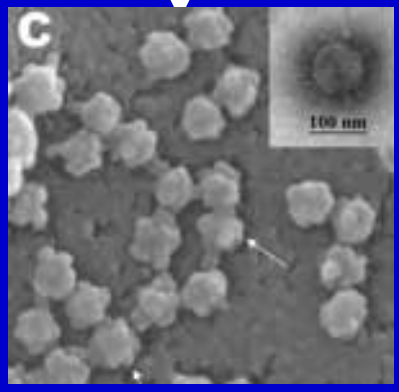
+15h

SARS infection (in the world)

SARS infection (in the cell)

4/30/03





+24h

Will the epidemic spread, or die out?

$X^2 + Y^2 = Z^2$

Solving equations

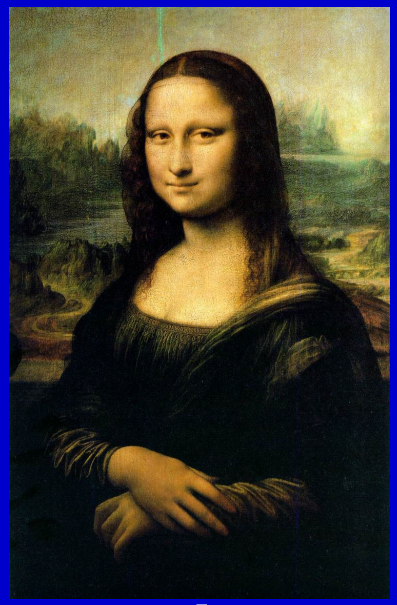
$X=3\ Y=4\ Z=5$

$X^n + Y^n = Z^n \quad n>2$

Proving theorems

Theorem: no solution!
Proof does not fit on this slide (200 pages)

Computations in Mathematics



Haiti earthquake: not enough doctors

BBC News - 9 hours ago

Between 100000 and 200000 people may have died in Haiti as a result of the devastating earthquake that struck last Tuesday.

Face recognition

Emotional reactions

"Mona Lisa"

Sadness

The subconscious brain computes

Beauty from computation

Seashells compute

How to describe computation?

The language of Algorithms

Father of Computing

Alan Turing 1912-1954

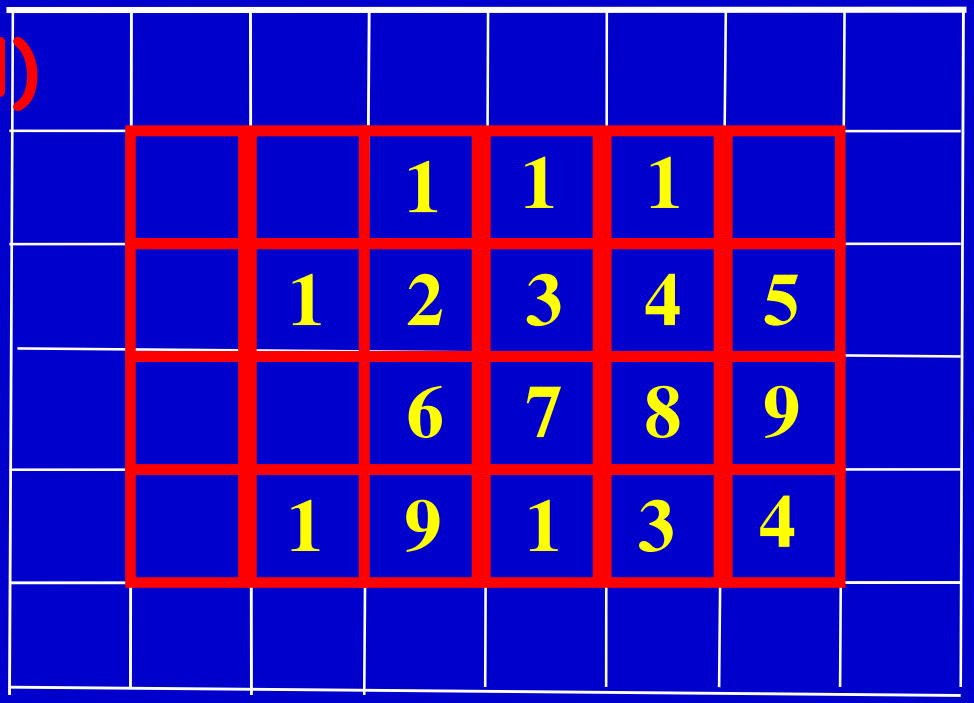
1936: "On computable numbers, with an application to the entscheidungsproblem"

- Formal definition of algorithm (Turing machine)
- Seed of the computer revolution
- Church-Turing Thesis: everything that nature computes, can be emulated on a Turing machine
- Limits on the power of algorithms.

ALGORITHM (informal)

Step-by-step, **local**, simple, mechanical procedure.

Halts in **finite** time for *every* input.



Example: Addition algorithm (informal)

1. Scan column. If empty, stop.
2. Add digits. Write answer, remember carry.
3. Move one column left, write carry.
4. Go to 1

Finite description vs. Infinite # inputs

Limits of Knowledge I

Unsolvable

Turing (& Godel): Given a computer program, does it always halt?

Mattiasevich: Given an equation, does it have an integer solution?

Conway: Given a (rule for) epidemic, will it spread or die?

Solvable

When?

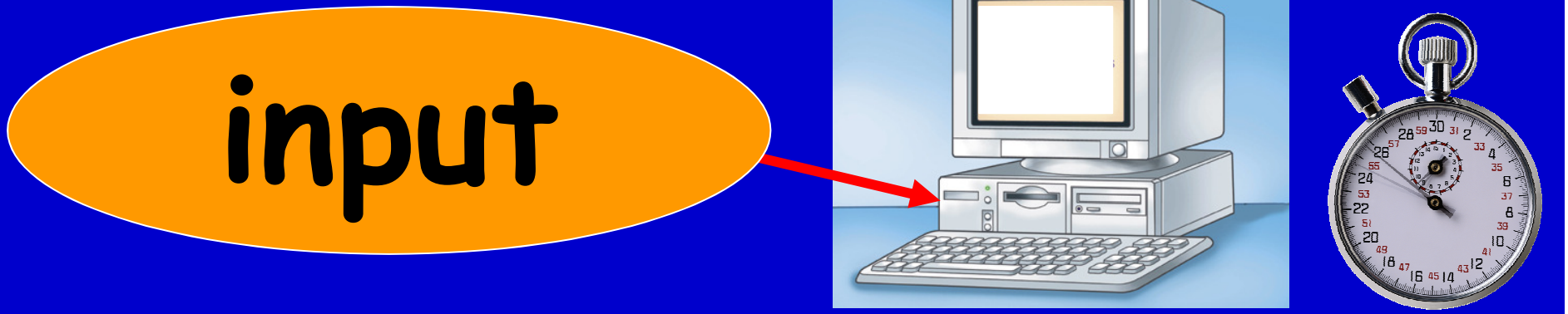
Computational Complexity Theory

Efficiency of an algorithm -

asymptotic analysis:

Number of basic steps,

for larger and larger inputs.



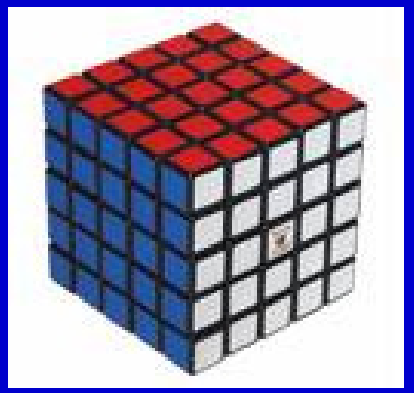
Rubik's cube

How many steps to solve..

2

3

4



5

...

Sudoku

How long does it take you to solve…

		8	6					
							6	
			4	8			2	3
		5		9				8
	4	9				2	1	
2				4		7		
3	6			2	9			
	1							
					5	1		

3

1			2	3	4			12		6				7	
		8				7			3			9	10	6	11
	12			10			1		13		11			14	
3			15	2			14				9			12	
13				8			10		12	2		1	15		
	11	7	6				16				15			5	13
			10		5	15			4		8			11	
16			5	9	12			1						8	
	2						13			12	5	8			3
	13			15		3			14	8		16			
5	8			1				2				13	9	15	
		12	4		6	16		13			7				5
	3			12				6			4	11			16
	7			16		5		14			1			2	
11	1	15	9			13			2				14		
	14				11		2			13	3	5			12

4

Sudoku

	y				b		a	c	x	n			h			t		f	l			d	e	
	t		s	u	j	h	v				d	q	c			o	k		b	n		a	w	p
w	h	e	m	a	n		l	u	k	p		r	y		s	x	d	q	c	o	j		i	b
b		j		p	s			t				i	m		v	n	g	h	a	q		r	x	y
x	o	l	d		i	p			r	e			f		u	j	w	y	m		h		s	c
	w	q	u	j			i		e		x	b	o	m	a			n	h	k	c	s		
n	c				w	x	u	s	f		q		l			e	m	k	v				j	a
a	i		x	f	c	l			m		v	k	w		q			j		d	g		b	h
s			v		h	k	p	o	b	u	f	j	n				t		d	i	m		r	q
	b	d		m	r	v			j		h	p			o	g	y	w			t		u	
y	p		e	l	a	m		v	h	o	b		x	i	t	s	q	u	w	g	r	c	d	k
	q	g	j		e		s	r		h	c				f	k			x		y	l	a	o
		u	t	k		n	o		l		r	m	q	y		b	a	v	j		i	p	h	
	x	r		w	p		y	k	i		l	e	j					m		t	q	v		u
	s		n	b	q	c		g	w	k	a	u	t	p	y		o		r	x		j	m	
j	n	s	q	v	x	y	h		u	t	p	o	g	l	m		f	d			w	i	k	r
u		w	b	t	l	e	r	p	o	m		c	d	f	k	v					s	q		
d			h		m	s	c	f		q	j		k	n	g	w		b		l	v	u		e
			o	e	d	i	k	n	q		w		u		j	a	l			h		b	p	m
l	k				v	j	t	w		a	s	h					u	r	q	c	d	f		n
			g	d	y	r	w			c		l	i		n	p	v	a	f	e			q	
	v	x	p	o		t	b			d	n	f			w			g		s	a	h	y	i
i		k	w	c	g	q	x	h					a	u	l	d	e		s			m	f	v
		a	y	r		d	f	e	n	x	k		s	h			b		u		p			
q	l		f	s			m	i	v			w			h		x	t	y				c	d

5

.....

Efficiency of the addition algorithm

1. Add digits. Write answer, retain carry.
2. Move one column left, write carry.
3. Scan column. If empty, stop.
4. Go to 1

6 basic steps per column

5 DIGI TS
30 STEPS

```
  12345
+ 6789
```

10 DIGI TS
60 STEPS

```
  123456789
+ 987654321
```

20 DIGI TS
120 STEPS

```
  72635273545786043726
+ 53827484732625435473
```

50 DIGI TS
300 STEPS

```
  47563739203487456438992305757328576452364568456465744576
+ 98656092843467546234868431987543210979832658741346534 72
```

N DIGITS
6N STEPS

Is there a faster algorithm? No!

Solving is as fast as reading the input

Efficiency of the multiplication algorithm

5 DIGI TS
25 STEPS

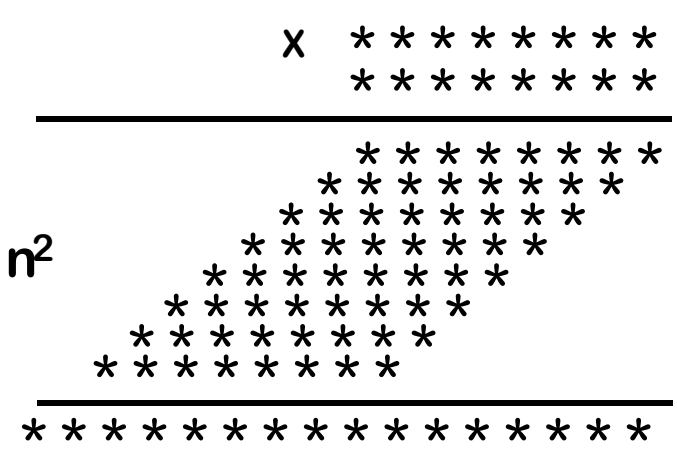
10 DIGI TS
100 STEPS

20 DIGI TS
400 STEPS

50 DIGI TS
2500 STEPS

N DIGITS
N^2 STEPS

Grade-school multiply algorithm



12345
x6789

123456789
x 987654321

72635273545786043726
x53827484732625435473

475637392034874564389923057573285764523645684564657445769865609284346754623486843198754321097983286587413465347 2

Fast! Multiply 10,000 digits in a second!

Is there a faster algorithm? Yes!

But not as fast as addition

Efficiency of a factoring algorithm

? × ? = 147,573,952,588,676,412,927

Find nontrivial factors of a number *A*

N DIGITS

$10^{N/2}$ STEPS

Brute force factoring algorithm

Input: A

- For B = 2,3,…√A do:
- If B divides A, return B, A/B

Very slow! 1000 digits → sun will die before finishing

Is there a faster algorithm?

Yes, but still extremely slow!

Which problems are hard to solve?

Addition & Multiplication: Easy

Is **Factoring** hard ?

Finding efficient algorithms, or
proving that no such algorithms exist:
Bread and butter of our field

Cobham, Edmonds
Rabin ~1965

The class P

All problems having an efficient (polynomial time, e.g. n, n^2) algorithm like Addition and Multiplication

Many practical interesting problems in P

Efficient algorithms –

Drivers of invention & industry

Who were

Edison ? Marconi ? Guttenberg ? Stevenson ?

Light bulb Radio Printing press Steam engine

Dijkstra ? Tukey ? Berlekamp ? Knuth ?

Inventors of important efficient algorithms

Shortest path

Dijkstra 1959

Google Maps

MAPQUEST

Network flows

Internet routing

Dynamic Programming

.....


```
define Dijkstra(Graph G, Node s)
        S := {}
        Q := Nodes(G)
        while not empty(Q)
            u := extractMin( Q )
            S := S ∪ u
            for each node v in neighbors( u )
                if d(u) + w(u,v) < d(v) then
                d(v) := d(u) + w(u,v)
                pi(v) := u
```

Distance (Delhi, Bangalore)

Path (Delhi, Bangalore)

Pattern matching

Knuth-Morris-Pratt

Boyer-Moore 1977

Spell checking

Text processing

Genome

Molecular Biology

Web search

Text CAUCGCGCUUCGC

Pattern CGC

```
algorithm kmp_search:
    input: T (text), P (pattern sought)
    define variables:
        m ← 0, i ← 0, M (the table)
    while m + i is less than length of T, do:
        if P[i] = T[m + i], let i ← i + 1
            if i = length of P then return m
        otherwise, let m ← m + i - M[i],
            if i > 0  let  i ← M[i]
```

Text CAUCGCGCUUCGC

Location X X X

Fast Fourier Transform (FFT)

Cooley-Tukey 1965

Gauss 1805

Audio processing

Image processing

Tomography, MRI

Fast multiplication

Quantum algorithms

T(0), T(1), T(2), ….T(N)

```
RECURSIVE-FFT(a)
 1  n ← length[a]
 2  if n = 1
 3     then return a
 4  ω_n ← e^(2πi/n)
 5  ω ← 1
 6  a^[0] ← (a_0, a_2, . . . , a_(n−2))
 7  a^[1] ← (a_1, a_3, . . . , a_(n−1))
 8  y^[0] ← RECURSIVE-FFT(a^[0])
 9  y^[1] ← RECURSIVE-FFT(a^[1])
10  for k ← 0 to n/2 − 1
11        do y_k ← y_k^[0] + ω y_k^[1]
12           y_(k+(n/2)) ← y_k^[0] − ω y_k^[1]
13           ω ← ω ω_n
14  return y
```

$$T_N(x) = \sum_{n=0}^{N} a_n \cos(nx) + \mathrm{i} \sum_{n=0}^{N} a_n \sin(nx)$$

Error correction

Reed-Solomon decoding

Petersen 60

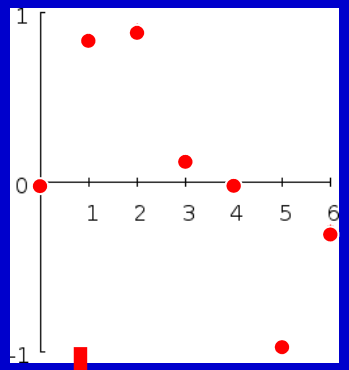
Berlekamp-Massey 68

CDs

DVDs

Satellite communication

Cell phone communication



INPUT: a binary sequence $S = S_O, S_1, S_2, \ldots S_n$.

OUTPUT: the complexity L (S) of S, $0 < L(S) < N$.

1. Initialization: C(D):=l, L:=O m:=-l, B{D):=l, N:=O.

2. While (N < n) do the following:

2.1 Compute the next discrepancy d.

$d := (S_N + \Sigma c_i S_{N-i}) \bmod 2$.

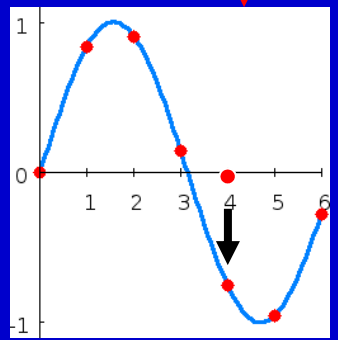
2.2 If d = 1 then do the following:

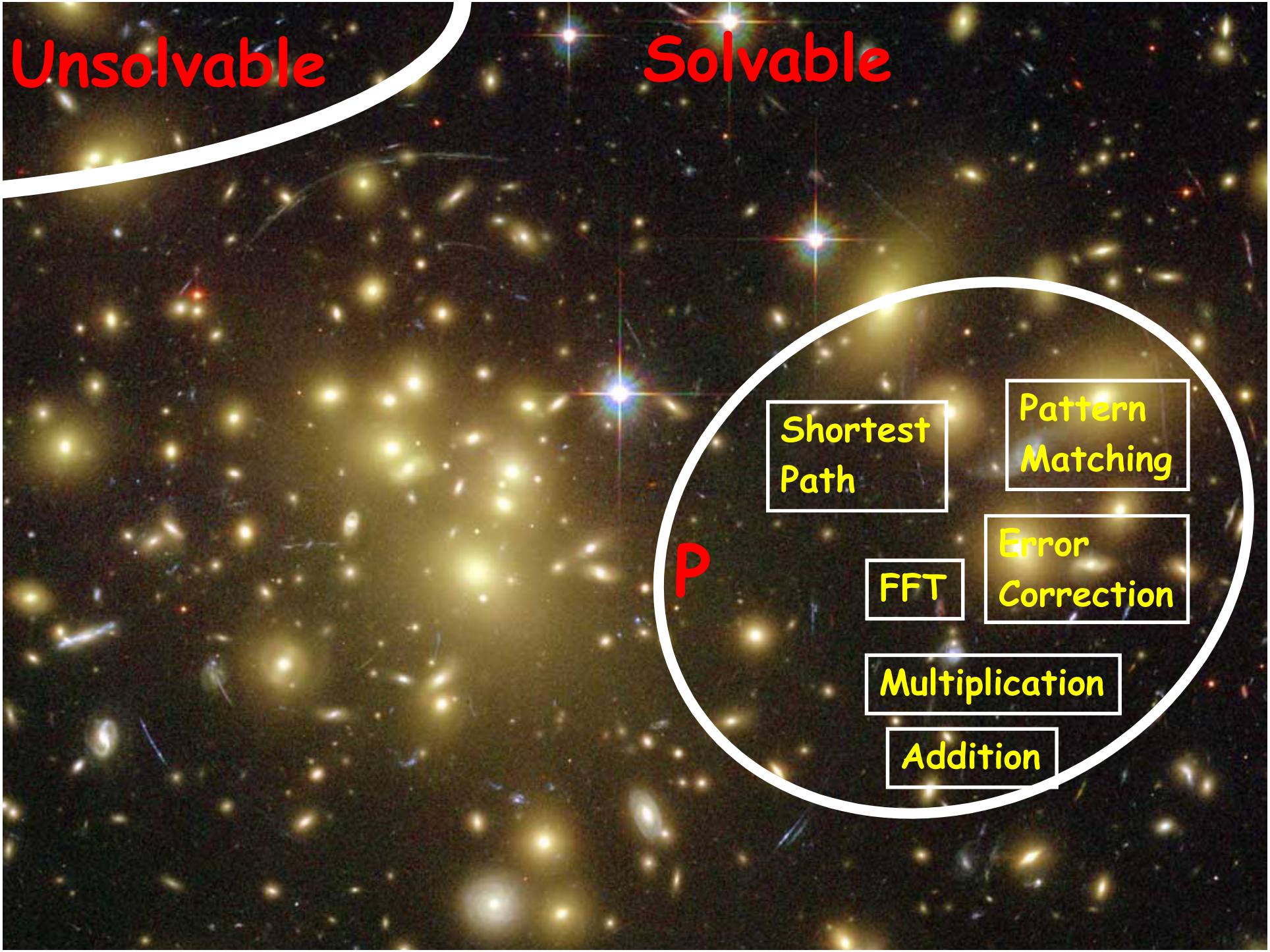
$T(D) := C(D)$, $C(D) := C(D) + B(D) \cdot D^{N-m}$.

If $L < N/2$ then $L := N+l-L$, $m := N$, $B(B) := T(D)$.

2.3 N:=N+l.

3. Return(L) .




Unsolvable
Solvable
P
Shortest Path
Pattern Matching
FFT
Error Correction
Multiplication
Addition

Cobham, Edmonds
Rabin ~1965

The class P

All problems having an efficient (polynomial time) algorithm

Many interesting problems in P

Are all interesting problems in P?
What are "interesting" problems?

Search problems

Short Path: FIND short path from Princeton to LA

Pattern Matching: FIND CGC in CAUCGCGCUUCGC

Easy!

Hard?

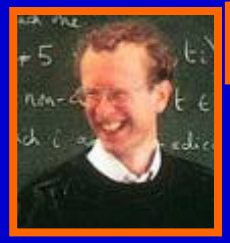
What is common to all these problems?

In all, solutions are easy to check & verify!

Factoring: FIND factors of 147,573,952,588,676,412,927

= 193,707,721 × 761,838,257,287

Theorem Proving: FIND a 200-page proof of the "Riemann Hypothesis"



Lemma…Proof…Lemma..Proof..

Sudoku: FIND solution of

		8	6					
							6	
			4	8			2	3
		5		9				8
	4	9				2	1	
2				4		7		
3	6			2	9			
	1							
					5	1		

9	2	8	6	1	3	4	5	7
4	7	3	9	5	2	8	6	1
1	5	6	4	8	7	9	2	3
7	3	5	2	9	1	6	4	8
6	4	9	7	3	8	2	1	5
2	8	1	5	4	6	7	3	9
3	6	7	1	2	9	5	8	4
5	1	2	8	7	4	3	9	6
8	9	4	3	6	5	1	7	2

The class NP- problems like FIND: needle in a haystack

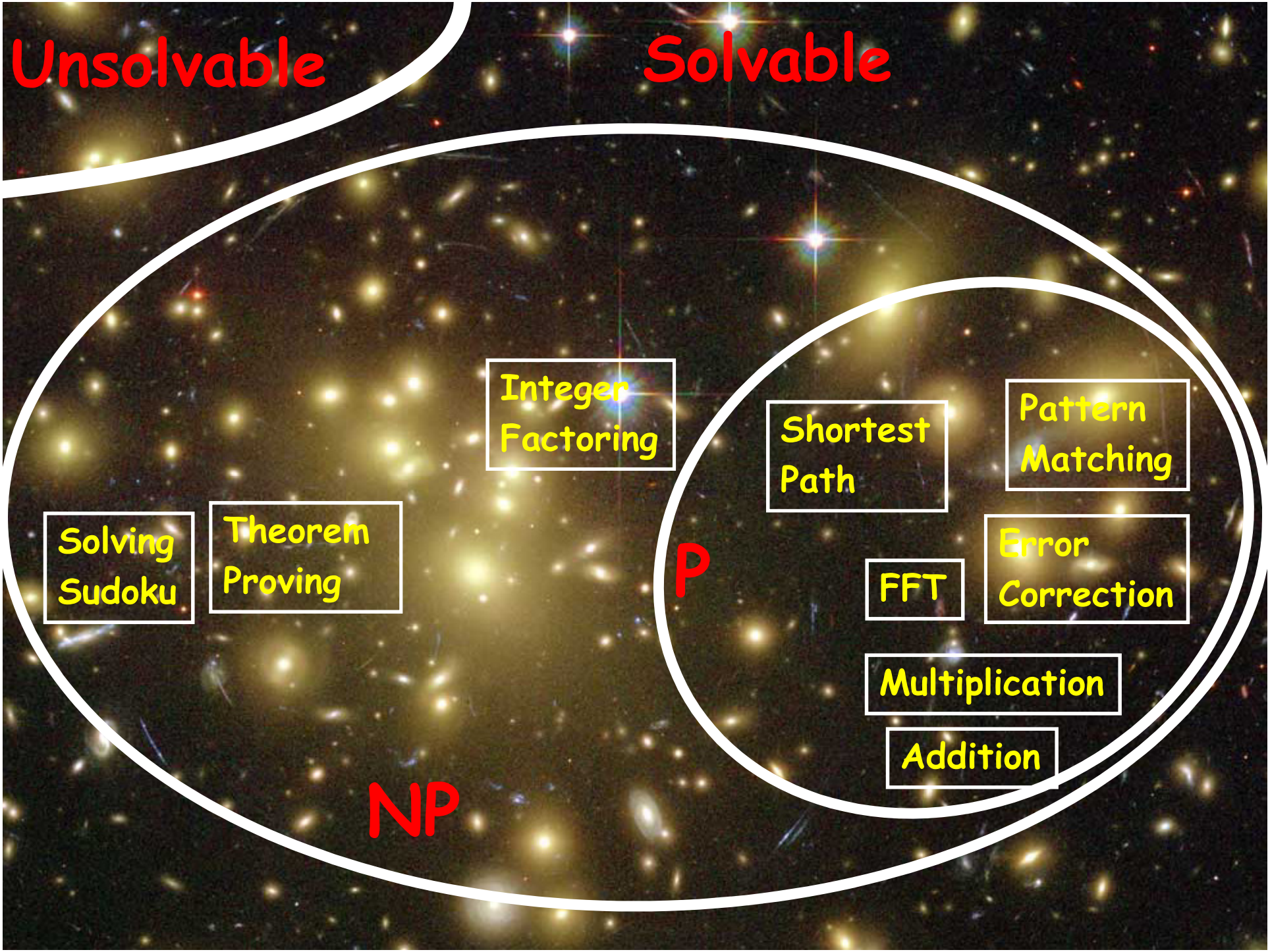
May be hard to *find*

Always easy to *verify*

Cook & Levin 1971
Gödel 1956

The class NP

All problems having efficient verification algorithms of given solutions


Unsolvable
Solvable
Integer Factoring
Shortest Path
Pattern Matching
Solving Sudoku
Theorem Proving
P
FFT
Error Correction
Multiplication
Addition
NP

Cook & Levin 1971
Gödel 1956

The class NP

All problems having efficient verification algorithms of given solutions

For every such problem, finding a solution (of length n) takes $\leq 2^n$ steps: try all possible solutions & verify each.

Can we do better than "brute force" ?

Do all NP problems have efficient algs ?

P versus NP

P: Problems for which solutions can be efficiently *found*

NP: Problems for which solutions can be efficiently *verified*

Conjecture: P ≠ NP

[finding is much harder than verification]

"P=NP?" is a central question of math, science & technology !!!

What is in NP?

Mathematician: Given a statement, *find* a proof

Scientist: Given data on some phenomena,
find a theory explaining it.

Engineer: Given constraints (size,weight,energy)
find a design (bridge, medicine, phone)

In many intellectual challenges, *verifying* that we found a good solution is an easy task !
(if not, we probably wouldn't start looking)

P=NP ➔ fast, automatic *finder*: Utopia!
"Creativity" can be efficiently automated!

Universality: NP-completeness

Are SuDoku, Theorem Proving, Factoring hard?

These problems are intimately related!!

Theorem: If SuDoku is easy then

- Theorem proving is easy
- Factoring is easy

Proof: SuDoku is NP-complete

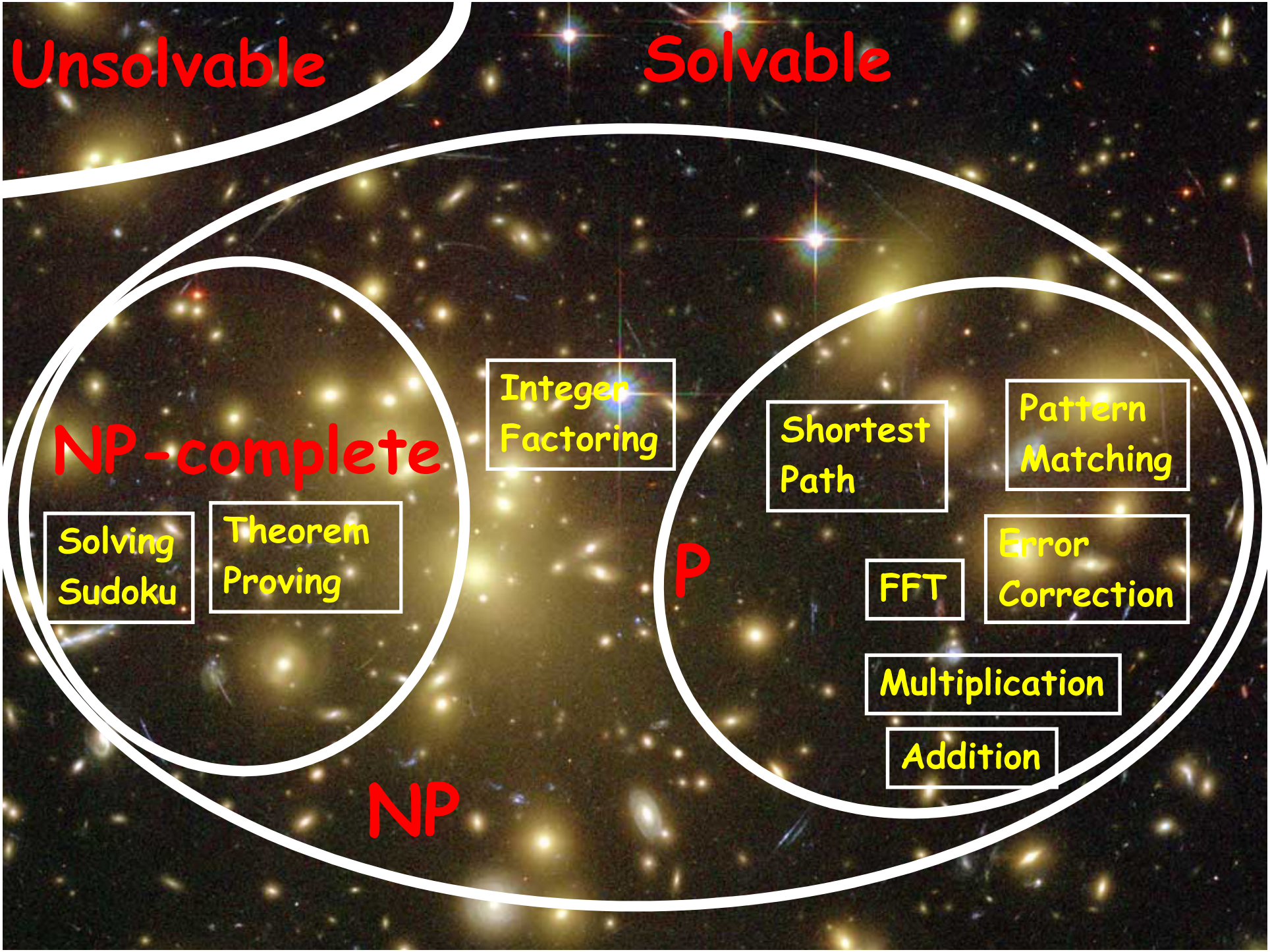
SuDoku solver can solve any NP problem

P=NP iff SuDoku has an efficient algorithm

Universality: NP-completeness

NP-complete problems:
If one is easy, then all are!
If one is hard, then all are!

SuDoku:	NP-complete
Thm proving:	NP-complete
Integer factoring:	we don't know


Unsolvable
Solvable
NP-complete
Solving Sudoku
Theorem Proving
Integer Factoring
NP
P
Shortest Path
Pattern Matching
FFT
Error Correction
Multiplication
Addition

Universality: NP-completeness

NP-complete problems:
If one is easy, then all are!
If one is hard, then all are!

SuDoku:	NP-complete
Thm proving:	NP-complete
Integer factoring:	we don't know

Thousands of NP-complete problems known in Math, Biology, Physics, Economics,….

Protein Engineering vol. 7 no. 9 pp. 1059-1068, 1994

The protein threading problem with sequence amino acid interaction preferences is NP-complete

Richard H. Lathrop

Economic Theory vol. 23, no. 2 , pp. 445-454, 2004

Finding a Nash equilibrium in spatial games is NP-complete

R. Baron, J. Durieu, H. Haller and P. Solal

[math.GR] arXiv:0802.3839v1

Quadratic equations over free groups are NP-complete

O. Kharlampovich, I.G. Lysenok, A G Myasnikov,N. Touikan

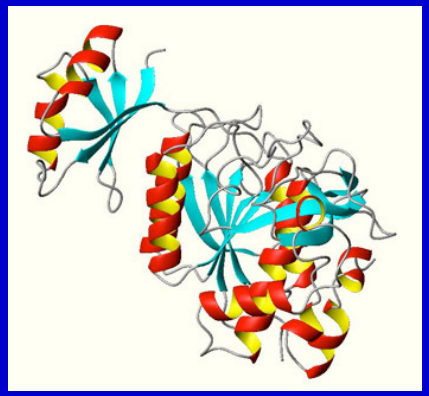
NP-completeness: sign of structural "nastiness".

Potential guide to better models and theories

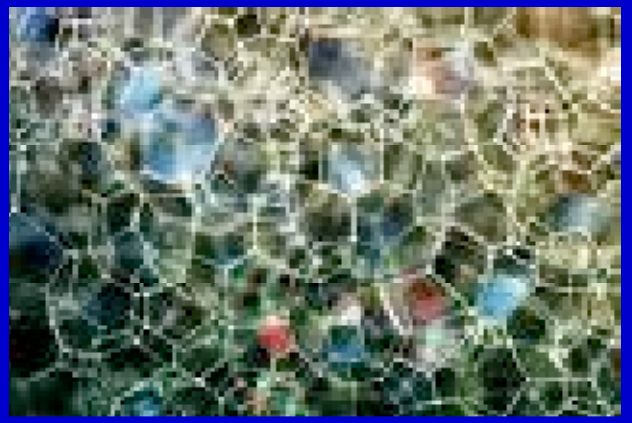
P ≠ NP as a law of nature

NP-complete problems that "nature solves"

Biology: Minimum energy Protein Folding



Physics: Minimum surface area Foam



Economics: Nash Equilibrium in strategic games

Possibilities:

model is wrong **or** inputs are special **or** P=NP

What is efficient computation?

Church-Turing Thesis:

Every *reasonable* process, can be efficiently? simulated by a Turing machine

- *Adding random bits*

Theorem [Blum-Micali, Yao, Nisan-Wigderson, Impagliazzo-Wigderson]

If "P≠NP", randomness add no power!

- *Adding quantum bits*

Theorem [Shor]

An efficient algorithm for Factoring

Positive consequences of P≠NP

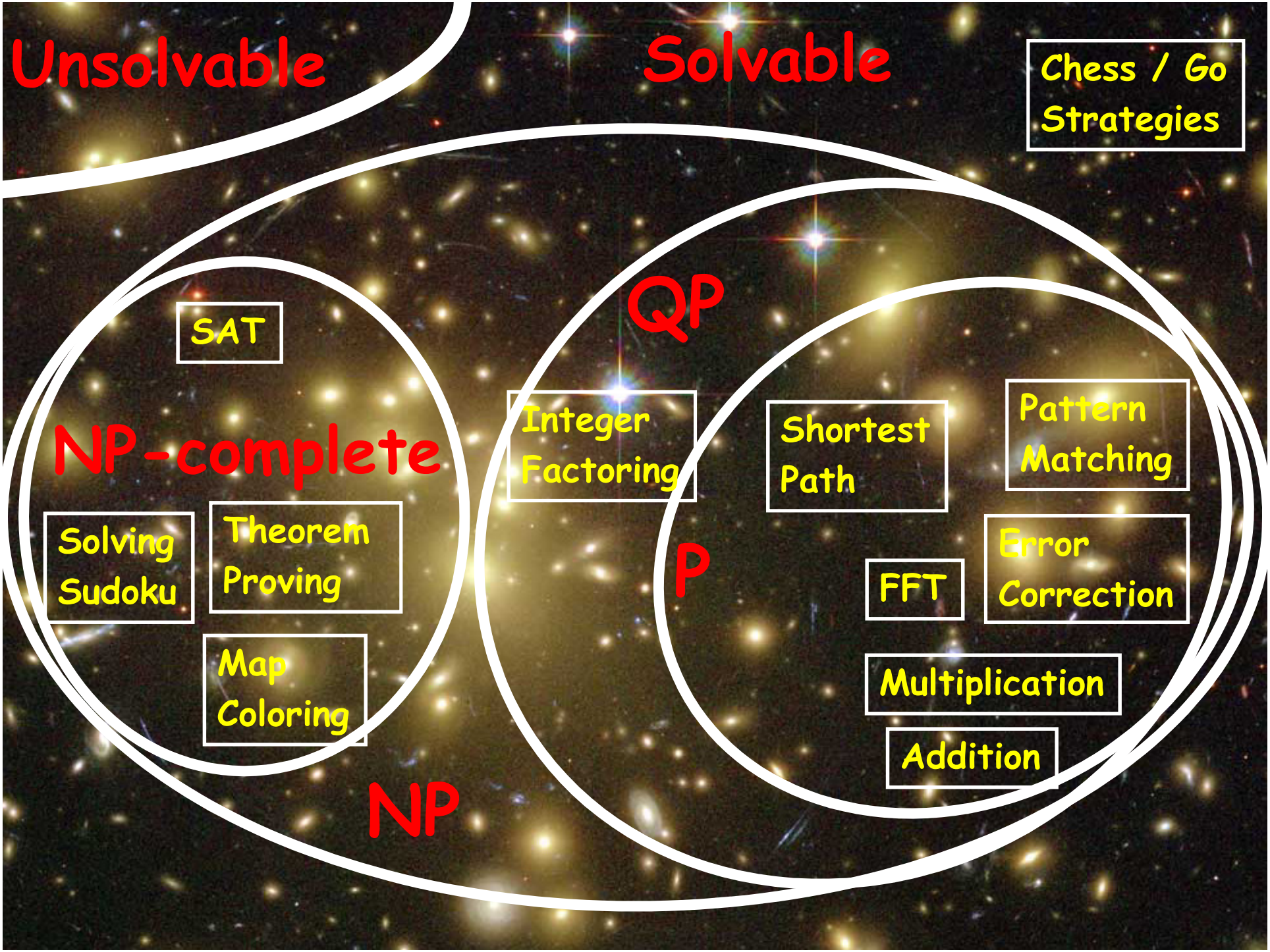
P≠NP Some of the problems we want to solve are hard. Are hard problems useful?

Cryptography: If **Factoring** is hard then:

- Encryption
- Digital signatures
- Secure e-mail
- Electronic commerce
- On-line shopping
- Poker by telephone

Things we didn't cover

- How to prove NP-completeness
- Attempts to prove $P \neq NP$ and restricted lower bounds
- Other resources (space, parallelism communication) and complexity classes
- Other modes of computation (average-case, approximate,…)
- ……


Unsolvable
Solvable
Chess / Go Strategies
SAT
NP-complete
Solving Sudoku
Theorem Proving
Map Coloring
NP
QP
Integer Factoring
P
Shortest Path
Pattern Matching
FFT
Error Correction
Multiplication
Addition