Lecture 1

Tuesday, May 9, 2017 1:09 PM

CMPT 407/710
Complexity Theory

Valentine Kabanets

kabanets@cs.sfu.ca www.cs.sfu.ca/~kabanets/407

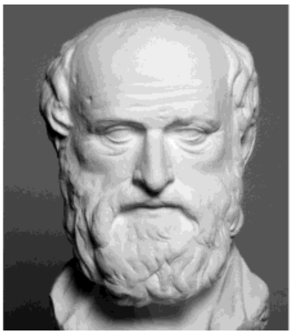
Four Algorithmic Problems

1. Is a given number N prime?
2. What is a greatest common divisor (GCD) of given numbers N and M?

3. Given a number N, factor it.

4. Given a polynomial eq'n (eg, $x^2+y^2=z^2$) does it have an integral solution?

1. Eratosthenes's Sieve



Write $1, 2, 3, 4, \ldots, N$.
Cross out all multiples of 2, then 3, then 5, ...

The survivors are primes.

Agrawal, Kayal, Saxena 2003:
Polytime Primality Testing!

2. Euclid's algorithm

GCD(M, N)

$r_0 := M$
$r_1 := N$
$i := 1$
while $r_i \neq 0$
  $r_{i+1} := r_{i-1}$ rem r_i
  $i := i+1$

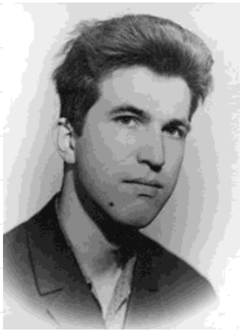
return r_{i-1}

3. Factoring not known in PolyTime

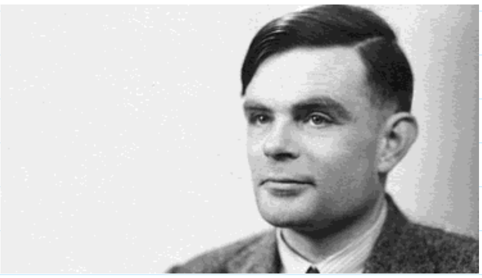
Peter Shor 1994:
Quantum PolyTime Algo
for Integer Factoring

RSA assumes Factoring is hard
(for security)!

4. Solving Diophantine Equations
is undecidable



Yuri Matiyasevich 1970
(based on [Robinson, Davis, Putnam])



Alan Turing:

- Turing machine
- undecidable problems exist!

Efficient Computation

ENIAC 1940's

1960s: P = Polytime Solvable
$\approx$ Efficient

Why P?

- Independent of the computer architecture.
- Can be defined without computers at all (say, using Logic).
- Captures many natural problems

Goal: Find a polytime algorithm for every given computational problem!

given computational problem?

Turing 1937: Some problems have no computer algorithms!

Time Hierarchy Thm (to be covered in this course):
For every time bound $t(n)$, there exist computational problems that require time $\geq t(n)$.
(n = input size)

Thus, Impossible & Hard problems exist!
Complexity Theory wants to understand the limits of efficient (polytime) algorithms.
Given a natural problem (e.g., Factoring), we wish to say
either: Easy (& here's a polytime algo.)
or: Hard (& here's a proof that no polytime algo. exists)

Complexity Theory = Computability Theory with Limited Resourses (time, space, ...)

Goals of Complexity Theory

- Understand what can & can't be <u>efficiently</u> computed.
- Taxonomy of problems according to required resources (time, space, ...)
- Study relations among complexity-theoretic concepts (algorithms, lower bounds, ...)

Some recent Successes

- Primality in P : [AKS 2004]
- ST-connectivity for undirected graphs in deterministic Logspace [Reingold '05]
- PCP Theorem : [AS '98, ALMSS '98]
- Hardness $\Rightarrow$ Pseudorandomness : [BM84, Y82, NW94, ...]

This Course

- What is known?
- What is conjectured?
- Open questions
- Ideas & algorithmic techniques behind complexity results.