The Power and Weakness of Randomness (when you are short on time)

Avi Wigderson
School of Mathematics
Institute for Advanced Study

Plan of the talk

- Computational complexity
  - -- efficient algorithms, hard and easy problems
- The power of randomness
  - -- in saving time
- The weakness of randomness
  - -- what is randomness ?
  - -- the hardness vs. randomness paradigm
- The power of randomness
  - -- in saving space
  - -- in distributed computing
  - -- to strengthen proofs

Easy and Hard Problems

a technology independent definition

Multiplication

mult(23,67) = 1541

grade school algorithm:
n^2 steps on n digit inputs

EASY

Factoring

factor(1541) = (23,67)

best known algorithm:
$\exp(\sqrt{n})$ steps on n digits

HARD?

-- we don't know!

-- the whole world thinks so!

Map Coloring and P vs. NP

Input: planar map M
(with n countries)

2-COL: is M 2-colorable? Easy

3-COL: is M 3-colorable? Hard?

4-COL: is M 4-colorable? Trivial

Theorem: If 3-COL is Easy
then Factoring is Easy

P vs. NP problem: Formal: Is 3-COL Easy?

Informal: Can creativity be automated?

Fundamental question #1

Is NP≠P ? More generally,
is any "natural" problem "hard"? E.g.

- Factoring
- 3-coloring
- Permanent
- Optimal Chess / Go strategies

Does NP (or even #P, or even PSPACE) require Exponential time/size ?

Public opinion: YES!

The Power of Randomness

Host of problems for which:

We have probabilistic polynomial time algorithms

We have no deterministic algorithms of subexponential time.

Coin Flips and Errors

Algorithms will make decisions using coin flips

0111011000010001110101010111...

(flips are independent and unbiased)

When using coin flips, we'll guarantee:

"task will be achieved, with probability >99%"

- We tolerate uncertainty in life
- Here we can reduce error arbitrarily <exp(-n)
- To compensate – we can do much more...

Number Theory: Primes

Problem 1: Given $x \in [2^n, 2^{n+1}]$, Is x prime?

NEW: Deterministic primality testing algorithm.

Problem 2: Given n, find a prime in $[2^n, 2^{n+1}]$

Algorithm: Pick at random $x_1, x_2, \ldots, x_{100n}$

For each x_i apply primality test.

Pr [$\exists i$ x_i prime] > .99

Algebra: Polynomial Identities

Is $\det(V(x_1, x_2,\dots, x_n)) - \Pi_{i<k} (x_i - x_k) \equiv 0$?

Theorem [Vandermonde]: YES

Given (implicitly, e.g. as a formula) a polynomial p of degree d. Is $p(x_1, x_2,\dots, x_n) \equiv 0$?

Algorithm: Pick r_i indep at random from {1,2,…,100d}

$p \equiv 0 \Rightarrow \Pr[\, p(r_1, r_2,\dots, r_n) = 0 \,] = 1$

$p \neq 0 \Rightarrow \Pr[\, p(r_1, r_2,\dots, r_n) \neq 0 \,] > .99$

Comments: Over small finite fields it is coNP-complete

Over large finite fields one can even factor p

Analysis: Fourier coefficients

Given (implicitely) a function $f:(Z_2)^n \rightarrow \{-1,1\}$ (e.g. as a formula), and $\varepsilon>0$,

Find all χ such that $|\langle f, \chi\rangle| \geq \varepsilon$

Comment : At most $1/\varepsilon^2$ such χ

Algorithm: ...adaptive sampling... Pr[success] > .99

Comment: Works for other Abelian groups.

Applications: Coding Theory, Complexity Theory

Geometry: Estimating Volumes

Given (implicitly) a convex body K in R^d (d large!)

(e.g. by a set of linear inequalities)

Estimate volume (K)

Comment: Computing volume(K) exactly is #P-complete

Algorithm:

Approx counting ≈ random sampling

Random walk inside K.

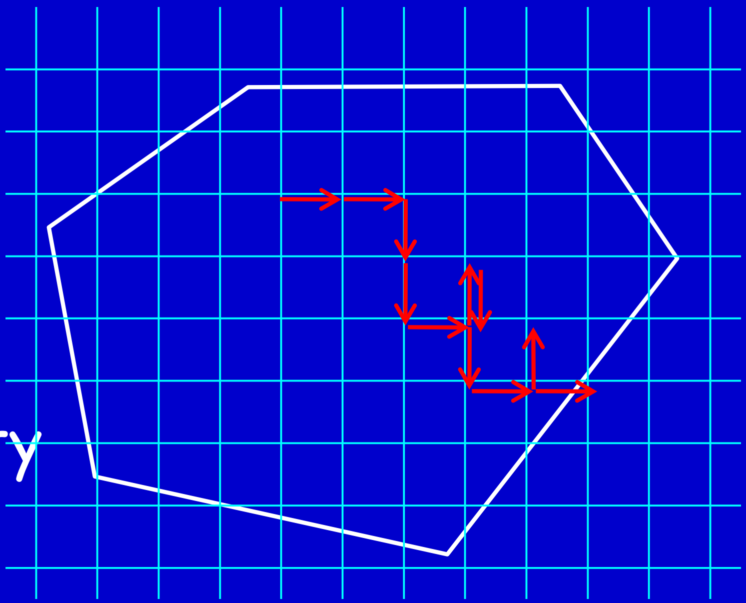
Rapidly mixing Markov chain.

Analysis:

Spectral gap ≈ isoperimetric inequality

Applications:

Statistical Mechanics, Group Theory



Fundamental question #2

Does randomness help?

Are there problems with probabilistic polytime algorithm but no deterministic one ?

Fundamental question #1

Does NP require exponential time/size ?

Public opinion: YES!

The public is WRONG on at least one question!

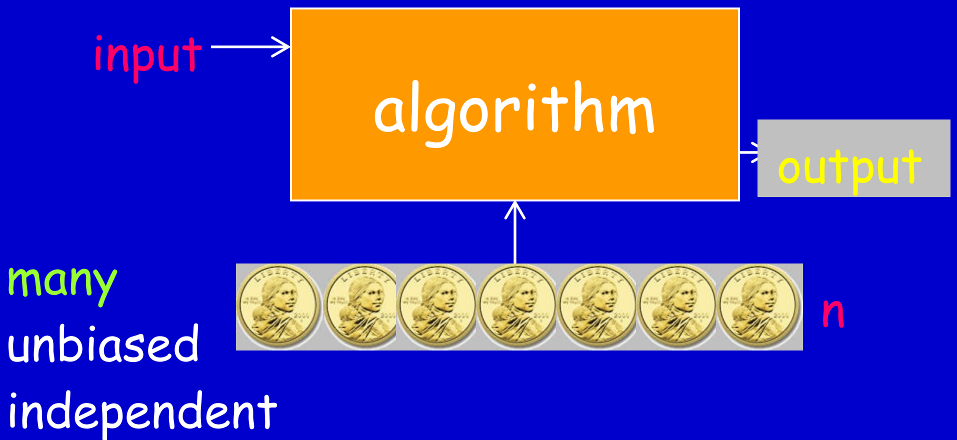
Hardness vs. Randomness

Theorem:

If there are natural hard problems (e.g. NP requires exponential size)

Then randomness does not save time (BPP=P)

Computational Pseudo-Randomness

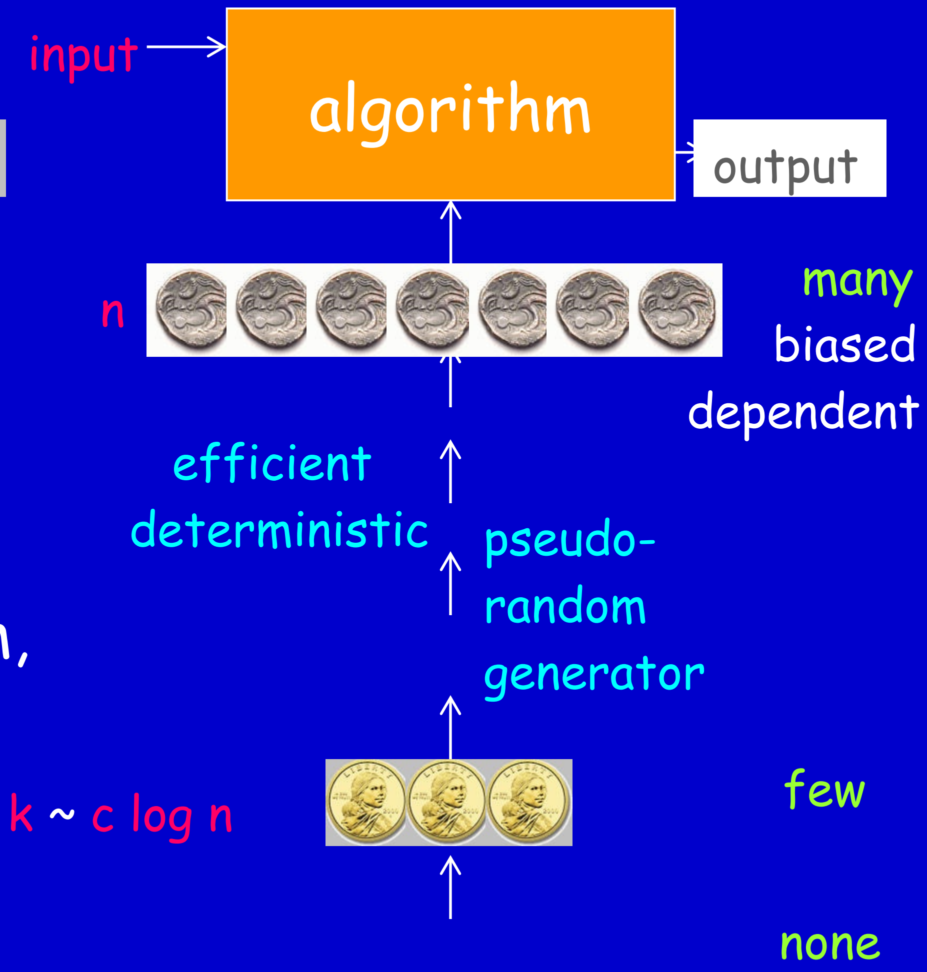


pseudorandom if

for every efficient algorithm,

for every input,

output ≈ output

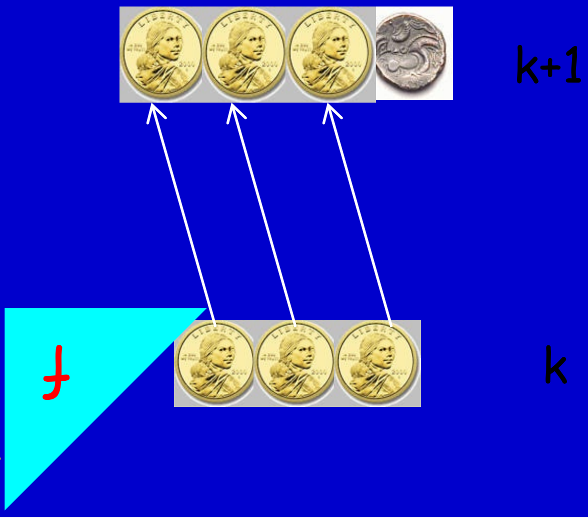


Hardness $\Rightarrow$ Pseudorandomness

$k \sim c \log n$

Want $G : \{0,1\}^k \rightarrow \{0,1\}^n$

We do $G : \{0,1\}^k \rightarrow \{0,1\}^{k+1}$



Need: Pr[$C(x) = f(x)$] < $1/2 + \exp(-k)$ for every computation C, size(C) < s — Average-case hardness

Hardness amplification

Have: Pr[$C'(x) = f'(x)$] < 1 for every computation C', size(C') < s' — Worst-case hardness

The Power of Randomness

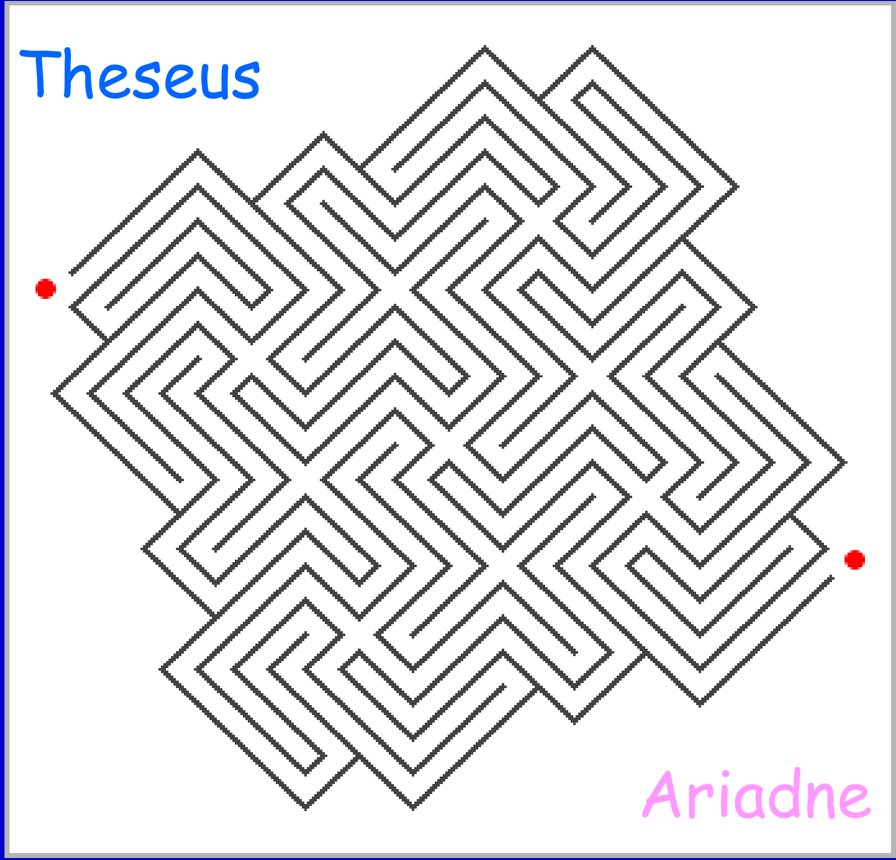
In other settings…

Getting out of mazes (when your memory is weak)

n–intersection maze

Only a local view

Theorem: A random walk will visit every intersection in n^2 steps (with probability >99%)

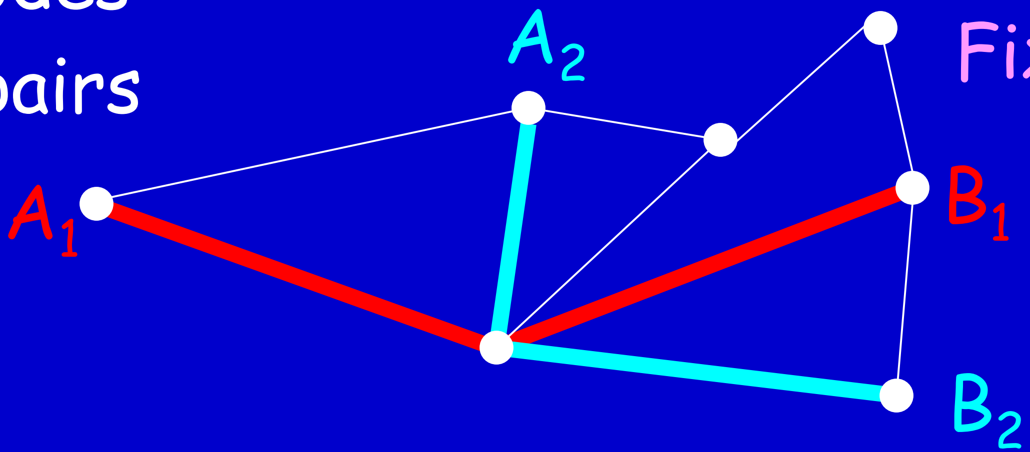


Crete, ~1000 BC

Decreasing Congestion in Networks

N nodes

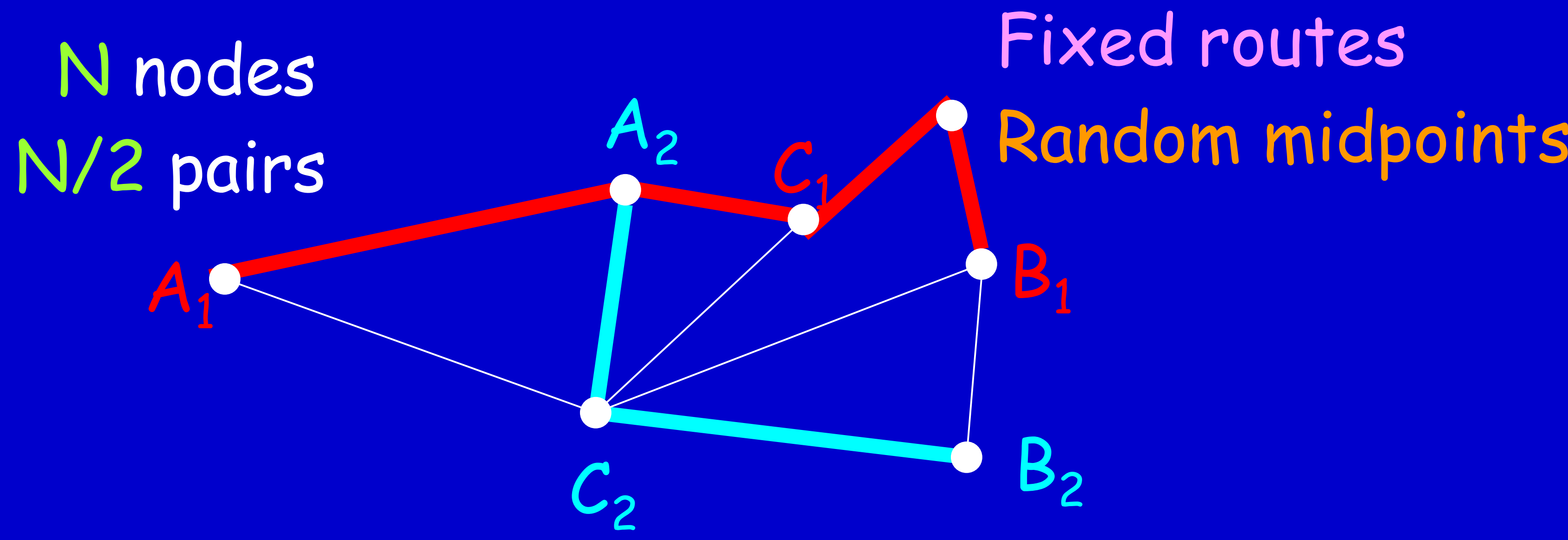
N/2 pairs



Fixed routes

Theorem 1: There is a choice of pairs (A_i, B_i) that will make a congestion of size $\sqrt{N}$ at some node

Decreasing Congestion in Networks



Theorem 2: If every pair (A_i,B_i) chooses a random intermediate point C_i, congestion drops to log N in all nodes (with probability 99%).

What is a Proof System?

Is a mathematical statement claim true? E.g.

claim: "No integers x, y, z, n>2 satisfy $x^n + y^n = z^n$"

claim: "The map of Africa is 3-colorable"

Prover

probabilistic

An efficient Verifier V(claim, argument) satisfies:

*) If claim is true then V(claim, argument) = TRUE for some argument always

(in which case claim=theorem, argument=proof)

**) If claim is false then V(claim, argument) = FALSE for every argument with probability > 99%

Remarkable properties of Probabilistic Proof Systems

claim: The Riemann Hypothesis

Prover: (argument)

Verifier: (editor/referee/amateur)

Probabilistically Checkable Proofs

Verifier's concern: Is the argument correct?

PCPs – refereeing (even by amateurs) in a jiffy!

Major application – approximation algorithms

Remarkable properties of Probabilistic Proof Systems

claim: The Riemann Hypothesis

Prover: (argument)

Verifier: (editor/referee/amateur)

Zero-Knowledge Proofs

Prover's concern: Will Verifier publish first?

ZK-proofs: argument reveals only correctness!

Major application - cryptography

Assumes: Factoring is HARD

Conclusions & Problems

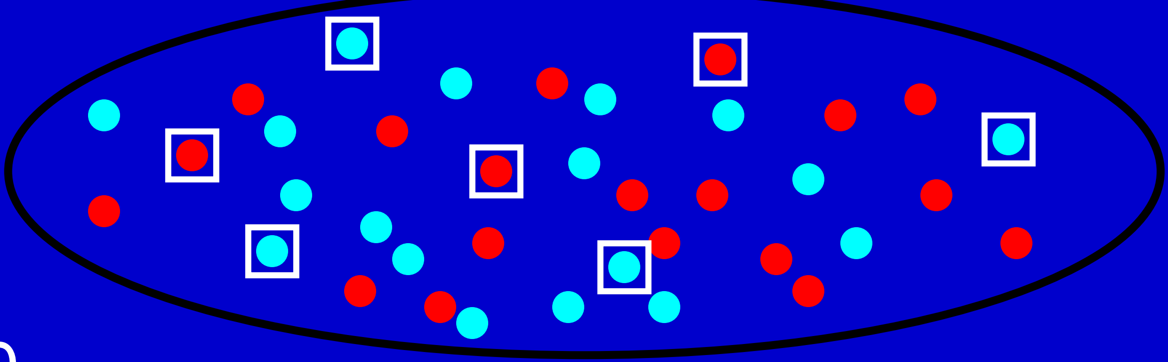
When resources are limited, basic notions get new meanings (randomness, learning, knowledge, proof, ...).

Randomness is in the eye of the beholder.
Hardness can generate (good enough) randomness.
Probabilistic algs seem very powerful but probably are not.
Sometimes this can be proven! (Small space algs,Primality)
Randomness is essential in some settings.

Is Factoring HARD? Is electronic commerce secure?
Is 3-COLOR HARD? Is P≠NP? Can creativity be automated?

Fast Information Acquisition

Population: 250 million, voting black or red



Random

Sample: 3,000

<u>Theorem</u>: With probability >99%

% in population = % in sample $\pm$ 5%

inependent of population size