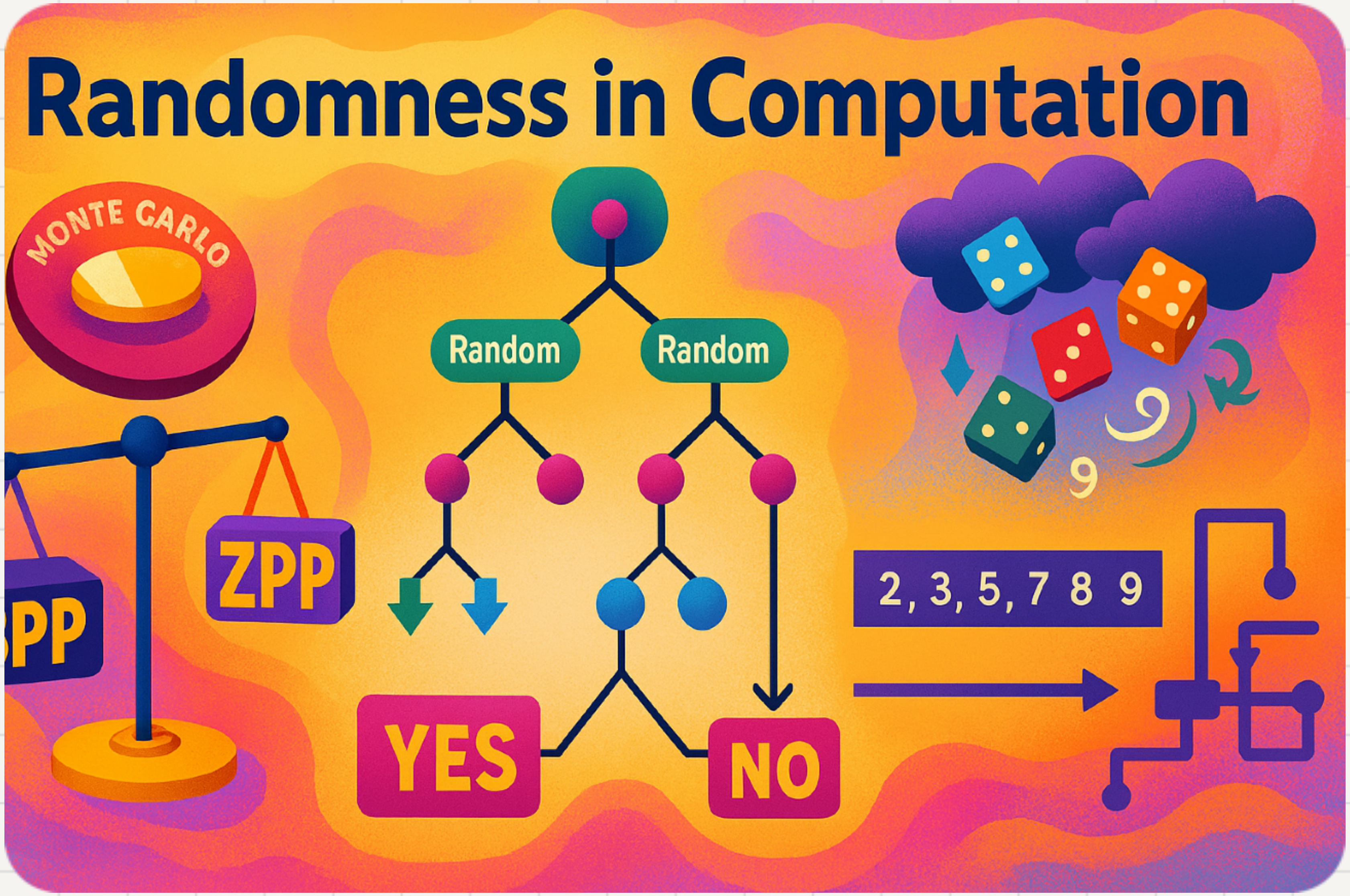
Complexity Theory



Randomness

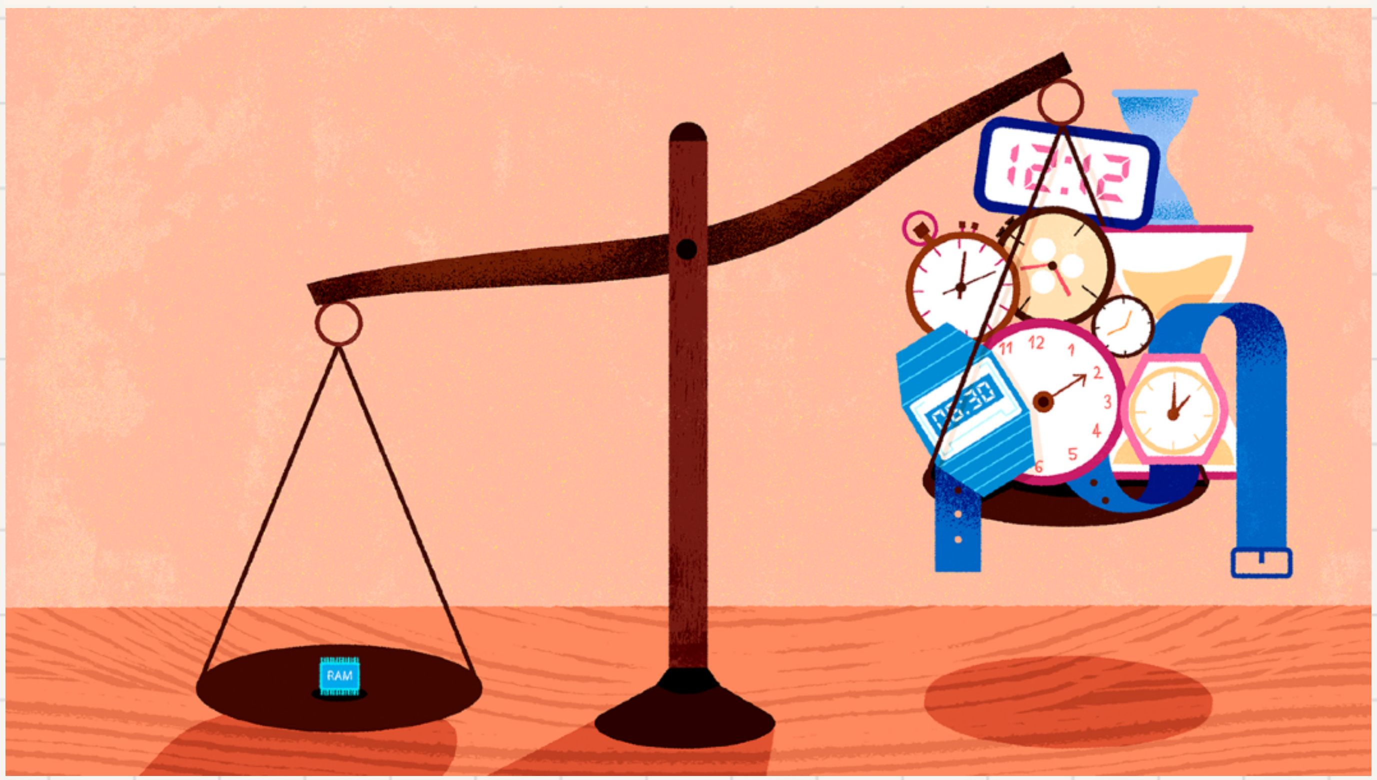
Last time: Time vs Space

COMPUTATIONAL COMPLEXITY

For Algorithms, a Little Memory Outweighs a Lot of Time

7

One computer scientist's "stunning" proof is the first progress in 50 years on one of the most famous questions in computer science.



Simulating Time With Square-Root Space*

Ryan Williams†
MIT

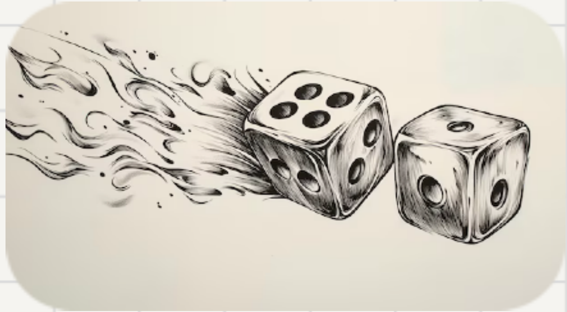
Abstract

We show that for all functions $t(n) \geq n$, every multitape Turing machine running in time t can be simulated in space only $O(\sqrt{t \log t})$. This is a substantial improvement over Hopcroft, Paul, and Valiant's simulation of time t in $O(t/\log t)$ space from 50 years ago [FOCS 1975, JACM 1977]. Among other results, our simulation implies that bounded fan-in circuits of size s can be evaluated on any input in only $\sqrt{s} \cdot \text{poly}(\log s)$ space, and that there are explicit problems solvable in $O(n)$ space which require $n^{2-\varepsilon}$ time on a multitape Turing machine for all $\varepsilon > 0$, thereby making a little progress on the P versus PSPACE problem.

Our simulation reduces the problem of simulating time-bounded multitape Turing machines to a series of implicitly-defined Tree Evaluation instances with nice parameters, leveraging the remarkable space-efficient algorithm for Tree Evaluation recently found by Cook and Mertz [STOC 2024].

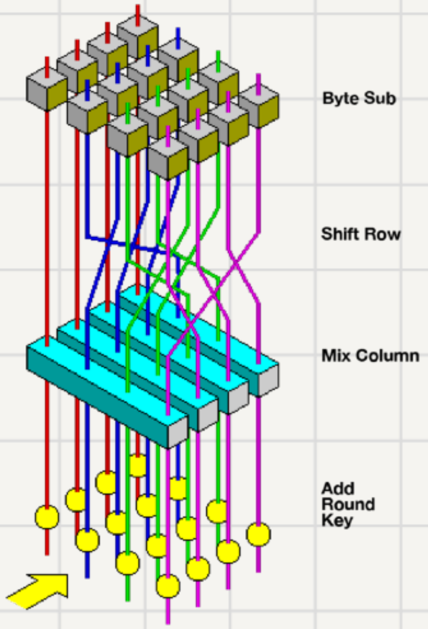
https://www.quantamagazine.org/for-algorithms-a-little-memory-outweighs-a-lot-of-time-20250521/

Conceptual questions we must ask



What is randomness?

Does randomness **exist**? Can free will exists in a deterministic world?



How do we **generate** randomness?

Is randomness in the **eye of the beholder**?

Probability or **uncertainty**?

How do we **model** randomness?

input

random string

Probabilistic Turing Machines $M(x; r)$

Primality testing

Problem: Given an integer $n \in \mathbb{N}$, is n a prime number?

Recall the naive algorithm:

Check divisibility by all $a < n$

Exponential complexity

Can randomness help? Idea—create a random test satisfying:

- If n a prime number, the test will always pass
- If n is **not** a prime number, the test will fail with high probability

Miller-Rabin Primality testing

Let $n \in \mathbb{N}$. Write $n - 1 = 2^s d$.

For $a \in [n]$, let $MR(n, a) = 1$ iff the following hold:

$$a^d \equiv 1 \pmod{n}$$

$$a^{2^r d} \equiv -1 \pmod{n} \quad \text{for some } r < s$$

Theorem
- If n a prime number, $MR(n, a) = 1$
- If n is **not** a prime number, $\Pr[MR(n, a) = 0] \geq 3/4$

What if 75% is not enough?

RP (Randomised Polynomial-time)

A language L is in RP if and only if there exists a Polynomial-time Probabilistic TM (PPT) M such that

- If $x \in L$, then $\Pr_r[M(x;r) = 1] \geq 1/2$
- If $x \notin L$, then $\Pr_r[M(x;r) = 0] = 1$

Where does Primality Testing lie?

Bonus: the class ZPP (Zero-error Polynomial-time Probablistic)

$ZPP \subseteq RP \cap coRP$

What's the catch?

RP Soundness amplification

A language L is in RP if and only if there exists a Polynomial-time Probabilistic TM (PPT) M such that

- If $x \in L$, then $\Pr_r[M(x;r) = 1] \geq 1/2$
- If $x \notin L$, then $\Pr_r[M(x;r) = 0] = 1$

Suppose that: • If $x \in L$, then $\Pr_r[M(x;r) = 1] \geq \epsilon$

Run the algorithm t times; output 1 iff one of the invocations accepted.

$$\Pr_r[M'(x;r) = 1] = 1 - \Pr_r[M'(x;r) = 0] \geq 1 - (1-\epsilon)^t$$

Choose $t = \log_{1-\epsilon}(\alpha)$ times, where α is the desired probability.

3SAT revisited

Consider a 3CNF formula ψ with the promise that:

- either **at least** 99% of the assignments satisfy ψ; or
- either **at most** 1% of the assignments satisfy ψ.

What is the complexity of deciding which is the case?

Now suppose:

- **more than 50%** of the assignments satisfy ψ; or
- **at most 50%** of the assignments satisfy ψ.

PP (Probabilistic Polynomial-time)

A language L is in PP if and only if there exists a Polynomial-time Probabilistic TM (PPT) M such that

- If $x \in L$, then $\Pr_r[M(x;r) = 1] > 1/2$
- If $x \notin L$, then $\Pr_r[M(x;r) = 1] \leq 1/2$

Is this the right notion?

Claim. $NP \subseteq PP$

Proof. Let $V(x, w)$ be the NP verifier of proof length $p(n)$

Consider the PP algorithm that on input $x \in \{0,1\}^n$:

1) Sample $w \in \{0,1\}^{p(n)}$ 2) If $V(x, w) = 1$ output 1 o/w flip a coin!

Note that $\Pr_w[V(x, w) = 1] > 1/2$

BPP (Bounded-error Probabilistic Polynomial-time)

A language L is in BPP if and only if there exists a Polynomial-time Probabilistic TM (PPT) M such that

- If $x \in L$, then $\Pr_r[M(x;r) = 1] \geq 2/3$
- If $x \notin L$, then $\Pr_r[M(x;r) = 1] < 1/3$

Can we generalise to:

- If $x \in L$, then $\Pr_r[M(x;r) = 1] \geq 1 - \epsilon$
- If $x \notin L$, then $\Pr_r[M(x;r) = 1] < \epsilon$

Why can't we use the RP amplification?

BPP Soundness amplification

Claim 1 (Chernoff bound). *Let $A_1, \ldots, A_t$ be independent identically distributed random variables taking values in $\{0,1\}$. Then,*

$$\Pr\left[\left|\frac{\Sigma_i A_i}{t} - \mathbb{E}[A_i]\right| \geq \delta\right] \leq 2e^{-t\delta^2/2} \,.$$

Let $A_1, \ldots, A_t$ be the outputs of invocations of $M(x)$.

Denote $A = \frac{1}{t}\sum_{i=1}^{t} A_i$ be the average output.

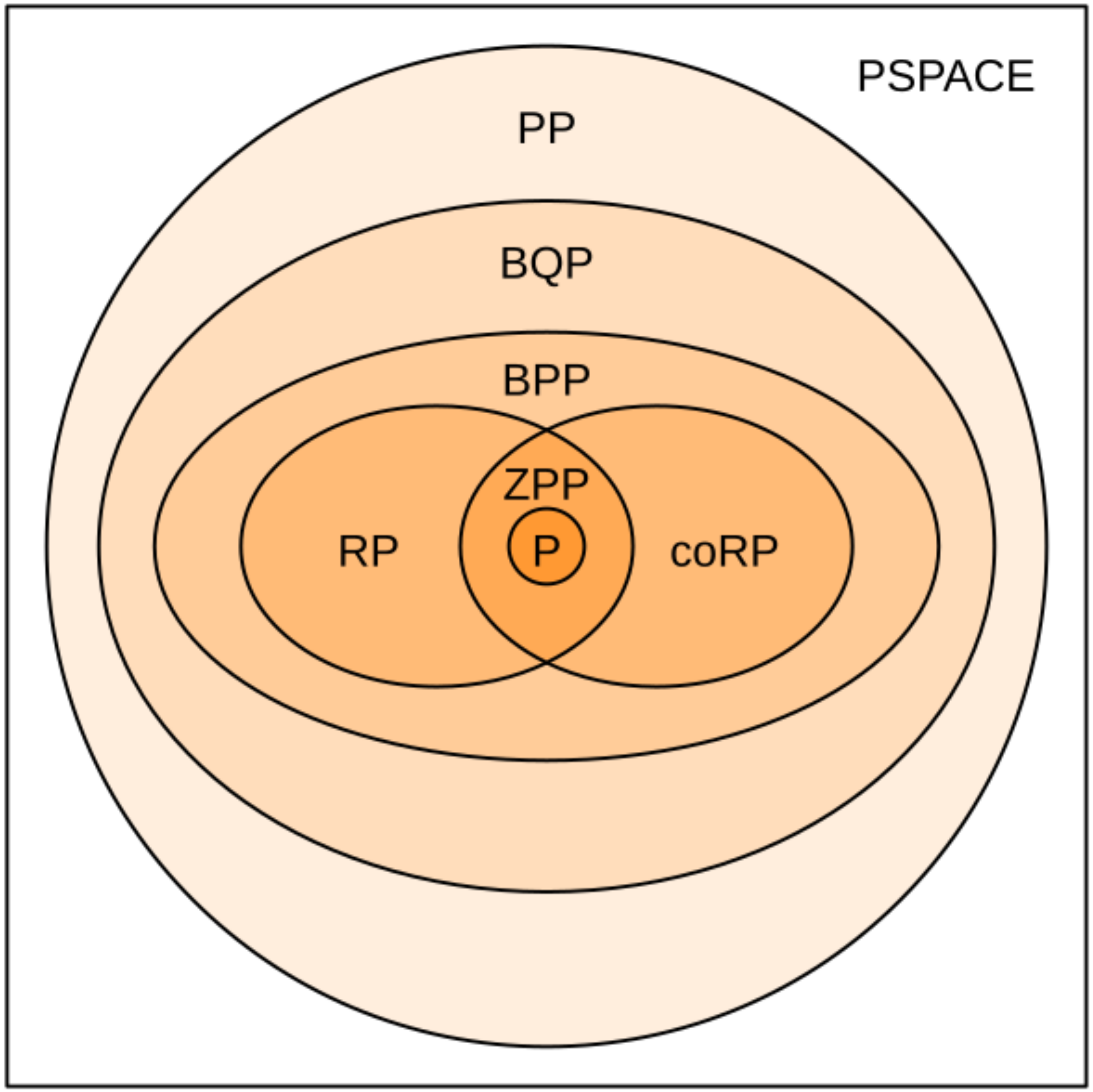
The amplified algorithm A' rules by majority.

On a 1-instance, $\mathbb{E}[A] \geq 2/3$, and the Chernoff bound gives

$$\Pr[\,|A - 2/3 \geq 1/6] \leq 2e^{-t(1/6)^2/2} = \exp(-t)$$

The analysis for 0-instances is symmetric.

Randomness complexity classes



Turing fact of the day

Pseudorandomness and **derandomisation** can be traced back to Turing!