Time Complexity
Computational Complexity Theory

What can and can’t be computed with limited resources on computation, such as time, space, randomness, communication, power,...

Captures many of the significant issues in practical problem solving.

The field is rich with important open questions, many are way beyond our current understanding.

We’ll start with: Time complexity
Measuring Time Complexity of a TM

We measure time complexity by counting the steps taken for a Turing machine to halt.

Consider the language $A = \{ 0^k1^k \mid k \geq 0 \}$

Here's a TM for $A$. On input of length $n$:

1. \textit{Scan} across the tape and reject if the string is not of the form $0^i1^j$.

2. \textit{Repeat} the following if both 0s and 1s remain on the tape:
   
   \textit{Scan} across the tape, crossing off a single 0 and a single 1.

3. If 0s remain after all 1s have been crossed off, or vice-versa, reject. Otherwise accept.
Let $M$ be a TM that halts on all inputs. (We will only consider *decidable languages* now!)

**Definition:**
The running time or time complexity of $M$ is the function $T : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$T(n) = \text{ maximum number of steps taken by } M \text{ over all inputs of length } n$$
Time-Bounded Complexity Classes

Definition:

\[ \text{TIME}(t(n)) = \{ L' | \text{there is a Turing machine } M \text{ with time complexity } \mathcal{O}(t(n)) \text{ so that } L' = L(M) \} = \{ L' | L' \text{ is a language decided by a Turing machine with } \mathcal{O}(t(n)) \text{ running time} \} \]

We just showed: \( A = \{ 0^k1^k | k \geq 0 \} \in \text{TIME}(n^2) \)
$$A = \{ 0^k1^k \mid k \geq 0 \} \in \text{TIME}(n \log n)$$

$$M(w) := \text{If } w \text{ is not of the form } 0^*1^*, \text{ reject.}$$

- Repeat until all bits of $w$ are crossed out:
  - If (parity of 0’s) $\neq$ (parity of 1’s), reject.
  - Cross out every other 0. Cross out every other 1.

- Once all bits are crossed out, accept.

```
0000000000000111111111111
x0x0x0x0x0x0x0xx1xlxlxlxlxlxl
xxx0xxx0xxx0xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1xxxx1 xxxx
It can be proved that a (one-tape) Turing Machine cannot decide A in less than $O(n \log n)$ time!

In fact:

Let $f(n) = O\left(\frac{n \log n}{\alpha(n)}\right)$ where $\alpha(n)$ is unbounded.

Prove: $\text{TIME}(f(n))$ contains only regular languages

For example, $\text{TIME}(n \log \log n)$ contains only regular languages
Theorem: \( A = \{ 0^k1^k \mid k \geq 0 \} \) can be decided in \( O(n) \) time with a two-tape TM.

Proof Idea:
Scan all 0s, copy them to the second tape. Scan all 1s. For each 1 scanned, cross off a 0 from the second tape.

Different models of computation can yield different running times for the same language

For \( B = \{ ww \mid w \in \{0,1\}^* \} \) the gap is quadratic
Theorem: Let \( t : \mathbb{N} \rightarrow \mathbb{N} \) satisfy \( t(n) \geq n \), for all \( n \). Then every \( t(n) \) time multi-tape TM has an equivalent \( O(t(n)^2) \) time one-tape TM.

Our simulation of multitape TMs by one-tape TMs achieves this!

Corollary: Suppose language \( A \) can be decided by a multi-tape TM in \( p(n) \) steps, for some polynomial \( p \). Then \( A \) can be decided by a one-tape TM in \( q(n) \) steps, for some polynomial \( q(n) \).
Theorem: Let $t : \mathbb{N} \to \mathbb{N}$ satisfy $t(n) \geq n$, for all $n$. Then every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time one-tape TM.
Theorem: Let $t : \mathbb{N} \rightarrow \mathbb{N}$ satisfy $t(n) \geq n$, for all $n$. Then every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time one-tape TM.
Theorem: Let $t : \mathbb{N} \rightarrow \mathbb{N}$ satisfy $t(n) \geq n$, for all $n$. Then every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time one-tape TM.
Theorem: Let $t : \mathbb{N} \rightarrow \mathbb{N}$ satisfy $t(n) \geq n$, for all $n$. Then every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time one-tape TM.
Theorem: Let $t : \mathbb{N} \rightarrow \mathbb{N}$ satisfy $t(n) \geq n$, for all $n$. Then every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time one-tape TM.
An Efficient Universal TM

**Theorem:** There is a (one-tape) Turing machine \( U \) which takes as input:
- the code of an arbitrary TM \( M \)
- an input string \( w \)
- and a string of \( t \) 1s, \( t > |w| \)
such that \( U(M, w, 1^t) \) halts in \( O(|M|^2 t^2) \) steps and \( U \) accepts \((M, w, 1^t) \) iff \( M \) accepts \( w \) in \( t \) steps

The Universal TM with a Clock

**Idea:** Make a multi-tape TM \( U' \) that does the above, and runs in \( O(|M| t) \) steps
The Time Hierarchy Theorem

Intuition: If you get more time to compute, then you can solve strictly more problems.

Theorem: For all “reasonable” \( f, g : \mathbb{N} \rightarrow \mathbb{N} \) where for all \( n, g(n) > n^2 f(n)^2 \), \( \text{TIME}(f(n)) \subsetneq \text{TIME}(g(n)) \)

Proof Idea: Diagonalization with a clock. Make a TM \( N \) that on input \( M \), simulates the TM \( M \) on input \( M \) for \( f(|M|) \) steps, then flips the answer.

Then, \( L(N) \) cannot have time complexity \( f(n) \)
The Time Hierarchy Theorem

Theorem: For all “reasonable” $f, g: \mathbb{N} \rightarrow \mathbb{N}$ where for all $n$, $g(n) > n^2 f(n)^2$, $\text{TIME}(f(n)) \subsetneq \text{TIME}(g(n))$

Proof Sketch: Define a TM $N$ as follows:

$$N(M) = \text{Compute } t = f(|M|). \text{ Run } U(M, M, 1^t) \text{ and output the opposite answer.}$$

Claim: $L(N)$ does not have time complexity $f(n)$.
Proof: Assume $N'$ runs in time $f(n)$, and $L(N') = L(N)$. By assumption, $N'(N')$ runs in $f(|N'|)$ time and outputs the opposite answer of $U(N', N', 1^{f(|N'|)})$

But by definition, $U(N', N', 1^{f(|N'|)})$ accepts $\Leftrightarrow N'(N')$ accepts in $f(|N'|)$ steps. This is a contradiction.
The Time Hierarchy Theorem

Theorem: For all “reasonable” $f, g : \mathbb{N} \rightarrow \mathbb{N}$ where for all $n$, $g(n) > n^2 f(n)^2$, \( \text{TIME}(f(n)) \subsetneq \text{TIME}(g(n)) \)

Proof Sketch: Define a TM $N$ as follows:

$N(M) = \text{Compute } t = f(|M|), \text{Run } U(M, M, 1^t) \text{ and output the opposite answer. So, } L(N) \text{ does not have time complexity } f(n)$. What do we need in order for $N$ to run in $O(g(n))$ time?

1. Compute $f(|M|)$ in $O(g(|M|))$ time [“reasonable”]
2. Simulate $U(M, M, 1^t)$ in $O(g(|M|))$ time

Recall: $U(M, w, 1^t)$ halts in $O(|M|^2 t^2)$ steps

Set $g(n)$ so that $g(|M|) > |M|^2 f(|M|)^2$ for all $n$. \text{ QED}

Remark: Time hierarchy also holds for multitape TMs
A Better Time Hierarchy Theorem

Theorem: For “reasonable” $f, g$ where
$g(n) \gg f(n) \log^2 f(n), \quad \text{TIME}(f(n)) \subset \text{TIME}(g(n))$

Corollary: \text{TIME}(n) \subset \text{TIME}(n^2) \subset \text{TIME}(n^3) \subset \ldots
There is an infinite hierarchy of increasingly more time-consuming problems

Question: Are there important everyday problems that are high up in this time hierarchy?

A natural problem that needs exactly $n^{10}$ time?
Polynomial Time

\[ P = \bigcup_{k \in \mathbb{N}} \text{TIME}(n^k) \]
The EXTENDED Church-Turing Thesis (variant)

Everyone’s Intuitive Notion of Efficient Algorithms \(\subseteq\) Polynomial-Time Turing Machines

More generally: TM can simulate every “reasonable” model of computation with only polynomial increase in time

A controversial thesis! Potential counterexamples: quantum algorithms
Parting thoughts:
More time more power
Different models may defer in polynomial factors – Extended Church-Turing Thesis hypothesize this is the worse possible. Polynomial-time a robust notion of efficiency (models, reductions ...)