LECTURE 1:
INTRODUCTION
Note: by Henri Mertens.
What is an algorithm?

A finite set of precise instructions acting on a finite input delivering a finite output halting in finite time, using only a finite amount of resources.
(1940s) Turing Machine

- Contains input, program, output

- Head: can read 1 bit, move tape left/right

Finite Automaton
RAM (RANDOM ACCESS MEMORY) as a model

CPU (central processing unit)

addresses

1 2 3 4

access any address in constant time

one cell can hold anything

a register holds an address, or data

MEMORY
POINTER-BASED MODEL

Register

CPU

access cells

(create no computations)

on addresses

(unlimited number of cells)

create, destroy, copy, fetch

nodes in constant time

node or cell

(holds data)

(has an address)
MEASURING TIME/COMPLEXITY

RAM model / Uniform cost measure.

All "operations" cost one time unit.

Bit model / Logarithmic cost measure.

Every bit operation costs one time unit.

Ex: Adding \( m \) bits and \( n \) bits costs about \( 2m \).

Can hold integers between \( 0 \) and \( 2^m - 1 \).
Each oracle access costs one time unit.

\[ E_x : \text{scale} \]

Answers: Is \( x < y \)?

\[ E_x : x \rightarrow \text{element} \rightarrow \text{set} \]

Answer the question: Is \( x \in A \)?

\[ E_x : x \rightarrow f(x) \]
OTHER MODES:

(cache)

CPU fast Cache Memory

(communication complexity)

Communicating 1 bit costs 1

Computer (internal computation costs 0)
LANDAU SYMBOLS

\( (a_n)_{n=1}^\infty, (b_n)_{n=1}^\infty \) are \( \geq 0 \) sequences.

\( a_n = O(b_n) \) if \( \exists m_0, c > 0 \) such that \( a_n \leq c \cdot b_n, \forall n \geq m_0. \)

\( a_n = \Theta(b_n) \) if \( \exists m_0, c_1, c_2 > 0 \) such that \( c_1 b_n \leq a_n \leq c_2 b_n, \forall n \geq m_0. \)

\( a_n = o(b_n) \) if \( \lim_{n \to \infty} \frac{a_n}{b_n} = 0. \) (also: \( a_n \ll b_n \))

\( a_n = \omega(b_n) \) if \( \lim_{n \to \infty} \frac{a_n}{b_n} = \infty. \) (also: \( a_n \gg b_n \))

\( a_n \sim b_n \) if \( \lim_{n \to \infty} \frac{a_n}{b_n} = 1. \)
EXAMPLES

\( A_n = 5n^4 + 3n^4 - 200n^4 \quad : \quad A_n = \Theta(n^4) \)

\( A_n \leq 8n^7 \quad \text{for all} \quad n \)

\( A_n \geq 5n^7 - 200n^6 \geq n^7 \quad \text{if} \quad n \geq 50 \)
**Time/Complexity**

\[ x_1, \ldots, x_m \rightarrow \text{Algorithm} \]

- **Worst-Case Time**
  \[
  T_m = \max_{x_1, \ldots, x_m} T(x_1, \ldots, x_m)
  \]

- **Average-Case Time**
  \[
  T_m = \frac{1}{\#(x_1, \ldots, x_m)} \sum_{(x_1, \ldots, x_m)} T(x_1, \ldots, x_m)
  \]

  \# of possible inputs
THE FIBONACCI EXAMPLE

0 1 1 2 3 5 8 13 21 34 55...

Compute $x_n$ when $x_0 = 0$, $x_1 = 1$, $x_n = x_{n-1} + x_{n-2}$, $n \geq 2$.

The mathematician's solution

$$x_n = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{\sqrt{5}}$$

If $x_n = x_m$, then need $x^2 = x + 1$, so $x = \frac{1+\sqrt{5}}{2}$ or $\frac{1-\sqrt{5}}{2}$. All solutions are given by $y \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + z \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n$.

Determine $y, z$ from $x_0 = 0$, $x_1 = 1$. 
A recursive program

\[ x(m) = \begin{cases} 
  m = 0 : & \text{return } 0 \\
  m = 1 : & \text{return } 1 \\
  m > 1 : & \text{return } x(m-1) + x(m-2) 
\end{cases} \]
\[ T_n = \text{time to compute } z(m) \text{ in RAM model} \]

\[ T_0 = T_1 = 1 \]

\[ T_m = T_{m-1} + T_{m-2} + 1 \quad \text{for } m \geq 2 \]

\[ (T_n + 1) = (T_{n-1} + 1) + (T_{n-2} + 1) \]

\[ (T_0 + 1) = 2 \]

\[ (T_1 + 1) = 2 \]

\[ m = 0 \ 1 \ 2 \ 3 \ 4 \]

\[ (T_n + 1) = 2 \times \text{Fibonacci} \]

\[ (n+1) = O\left(\left(\frac{1 + \sqrt{5}}{2}\right)^n\right) \]
Using an array

\[ x[0] = 0 \]
\[ x[1] = 1 \]

for \( i = 2 \) to \( m \) do

\[ x[i] = x[i-1] + x[i-2] \]

\[ f_m = H(m) \]
The matrix view:

\[
\begin{bmatrix}
\alpha_m \\
\alpha_{m-1}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
\alpha_{m-1} \\
\alpha_{m-2}
\end{bmatrix}
= \ldots
= \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\alpha_0
\end{bmatrix}
\]

How fast can we compute \( M^{m-1} \)?

RAM model: \( \mathcal{O}(\log m) \) time
Fast exponentiation

Recursive program \( \text{EXP} (M, n) \):

Case \( n = 1 \): return \( M \)

\( n > 1 \):
- \( m \leftarrow m \mod 2 \in \{0,1\} \)
- \( m \leftarrow m \div 2 \)
- return \( \left( \text{EXP} (M, m) \right)^2 \times M^0 \)

\[
M^m = \begin{cases} 
\left( M^{m/2} \right)^2 & \text{if } m \text{ even} \\
\left( M^{m/2} \right)^2 \times M & \text{if } m \text{ odd}
\end{cases}
\]

Why \( O \left( \log n \right) \)?

\( M^0 = \) unit matrix
COMPLEXITY CLASSES

- Constant: \( O(1) \)
- Logarithmic: \( \Theta(\log n) \)
- Polylogarithmic: \( \Theta(\log^k n) \)
- Linear: \( \Theta(n) \)
- Quadratic: \( \Theta(n^2) \)
- Polynomial: \( \Theta(n^k) \)
- Exponential: \( (1 + O(1))^n \)

\[ \log n \text{ to } \sqrt{n} \text{ to } n \text{ to } n^2 \text{ to } 2^n \text{ to } n! \]
There is life in between!