How to compare? How to choose?
- Speed
- Space
- Memory req.

How to compare speed?
- Run on a timer.

Select possible inputs and use them to time outputs

*benchmark, empirical testing.*

But *analytical testing*

- Represent each program as a mathematical object
- Use math to compare such objects,

Represent performance by a "runtime function".
Function of what? The input.
Factors
- Input size
- Input quality

"Runtime function" a function from input size to time,

$T(n)$ Comparing Programs.
\[ T_A(n) \quad \text{versus} \quad T_B(n) \]

\[ T_A(n) = 123n^2 + 70 \quad \quad T_B(n) = \frac{1}{6}n^3 + 2 \]

- NOT interested in comparing functions for a particular input size.

- Interested in what happens as the input becomes larger and larger \( \rightarrow \) "rate of growth" of functions.

**The Mathematics of the Growth of Functions.**

- **Big-O (Donald Knuth)**

**DEF:** Given two functions \( f(x) \) and \( g(x) \) we say that \( f(x) \) is \( O(g(x)) \) if there exist constants \( C \) and \( n_0 \) such that for every \( n > n_0 \)

\[ f(n) \leq C \cdot g(n) \]

**English Interpretation.**

- \( f(x) \) is \( O(g(x)) \) means that, ignoring constant factor, for sufficiently large values \( g(x) \) is larger or equal to \( f(x) \).
• $f(x)$ is $O(g(x))$ means that the rate-of-growth of $g(x)$ is greater than or equal to the rate-of-growth of $f(x)$

• $n^2$ is $O(3n^2 + n)$  \[ C = 1 \quad n_0 = 1 \]
  \[ n^2 \leq 3n^2 + n \quad \text{for any } n > 1 \]

• $3n^2 + n$ is $O(n^2)$
  \[ C = 4 \quad n_0 = 1 \]
  \[ 3n^2 + n \leq 4 \cdot n^2 \]
  \[ 3n^2 + n \leq 3n^2 + n^2 \]
  \[ n \leq n^2 \]

<<Ignoring Constant Factors>>

DEF Big-$\Theta$  

if $f(x)$ is $O(g(x))$ AND $g(x)$ is $O(f(x))$ then $f(x)$ is $\Theta(g(x))$  
and $g(x)$ is $\Theta(f(x))$

Big-$\Theta$ means $g(x)$ and $f(x)$ have the same rate-of-growth.

$3n^2 + 6n + 7$ is $O(9n^2 + 27n + 5)$ too cumbersome

• The complexity hierarchy:
  
  $n^3$ is $O(n^3)$
  $n^2$ is $O(n^2)$
  $n!$ is $O(n!)$
  $2^n$ is $O(2^n)$
  $\vdots$
  $n^2$ is $O(n^2)$

functions are not compared directly; instead, they are placed in hierarchy

- **Basic Rules:**

  R1) if \( T_1(x) \) is \( O(f(x)) \) and \( T_2(x) \) is \( O(g(x)) \) then
  \[
  T_1(x) + T_2(x) \leq O(f(x) + g(x))
  \]

  R2) if \( T_1(x) \) is \( O(f(x)) \) and \( T_2(x) \) is \( O(g(x)) \) then
  \[
  T_1(x) \cdot T_2(x) \leq O(f(x) \cdot g(x))
  \]

  R3) if \( T_1(x) \) is \( O(f(x)) \) and \( T_2(x) \) is \( O(g(x)) \) then
  \[
  T_1(x) + T_2(x) \leq O(\max(f(x), g(x)))
  \]

  \[
  2n \quad \text{is} \quad O(n) \quad \max(n, n^2)
  \]

  \[
  3n^2 \quad \text{is} \quad O(n^2) \quad 2n + 3n^2 \quad \text{is} \quad O(n^2)
  \]

  R4) A polynomial of degree \( k \) is \( O(n^k) \)