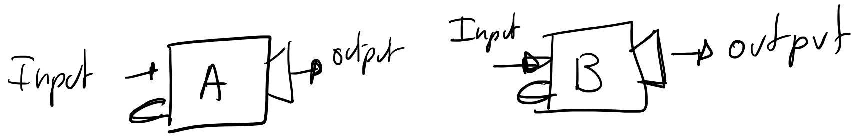
Intro to Algorithm Complexity

Wednesday, September 25, 2019 5:39 PM



How to compare? How to choose?

- Speed
- space
- memory req.

How to compare Speed.?

run on a timer.-

- run on same computer
- run on same environment
- compile on same compiler
  or
- implement on same language.

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 :- prefer size

Input quality.- assume worst quality

"runtime function" a function from input size to Time.

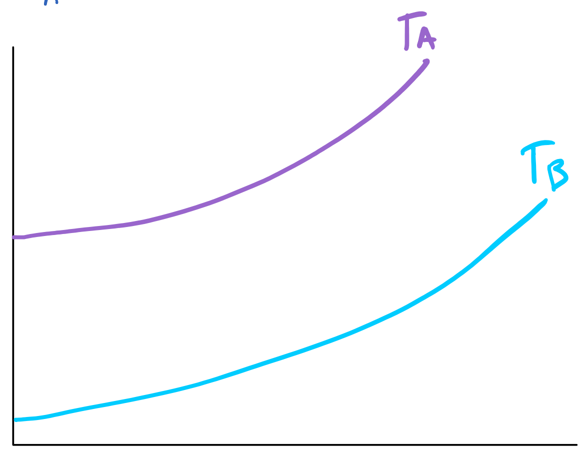
Comparing Programs.

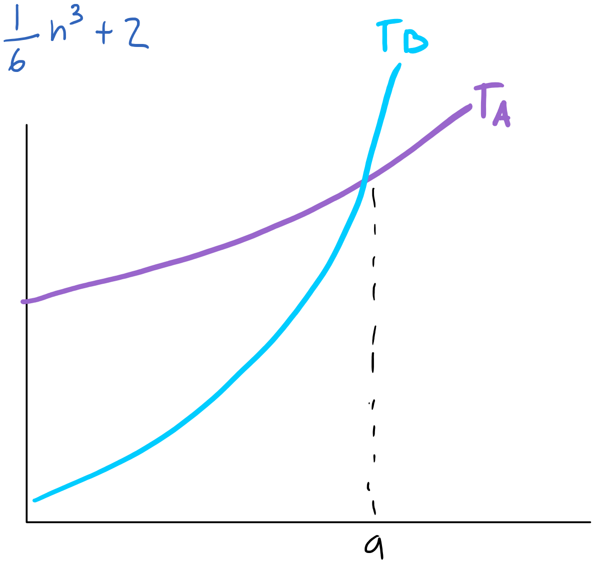
T( ) T(n)

$T_A(n)$ versus $T_B(n)$

Comparing Programs.
by
Comparing functions.

$T_A(n) = 123n^2 + 70$ $\qquad$ $T_B(n) = \frac{1}{6}n^3 + 2$





- NOT interested on comparing functions for a particular input size.
- Interested on what happens as the input becomes larger and larger
  ↳ "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.

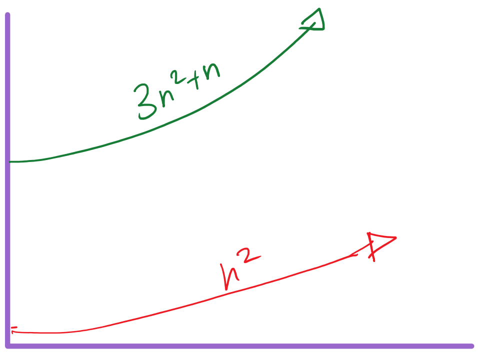
- $f(x)$ is $O(g(x))$ means that, ignoring constant factor for sufficiently large values $g(x)$ is larger or equal to $f(x)$.

- $f(x)$ 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 $O(3n^2+n)$ $\quad C = 1 \quad n_0 = 1$

$$n^2 \leq 3n^2 + n \quad \text{for any } n > 1$$

- $3n^2+n$ is $O(n^2)$ $\quad C = 4 \quad n_0 = 1$



$$3n^2+n \leq 4 \cdot n^2$$
$$\cancel{3n^2} + n \leq \cancel{3n^2} + n^2$$
$$n \leq n^2$$

《Ignoring constant factors》

DEF Big-Θ

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

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

n^2 is $O(n^3)$

n^3 is $O(n^2)$ ∅

- The complexity hierarchy:

$n!$
2^n
⋮
n^5
⋮

$\vdots$	
n^5	
n^4	
n^3	$\leftarrow \frac{1}{6}n^3 + 2$
n^2	$\leftarrow 3n^2+n \leftarrow 123n^2+70$ ✓
$n \cdot \log n$	
n	
$\log n$	
1	

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
$T_1(x) + T_2(x)$ is $O(f(x) + g(x))$

R2) if $T_1(x)$ is $O(f(x))$ and $T_2(x)$ is $O(g(x))$ then
$T_1(x) * T_2(x)$ is $O(f(x) * g(x))$

R3) If $T_1(x)$ is $O(f(x))$ and $T_2(x)$ is $O(g(x))$ then
$T_1(x) + T_2(x)$ is $O(\max(f(x), g(x)))$

$2n$ is $O(n)$

$3n^2$ is $O(n^2)$

$2n + 3n^2$ is $O(\underbrace{n^2}_{\max(n, n^2)})$

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