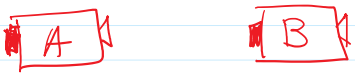
2 Introduction to the Analysis of Algorithms

Tuesday, August 29, 2023 11:48 AM

DEF: investigation of algorithm efficiency with respect to time and space.

Analytical, not empirical



which is better?

We will represent performance as a math function.

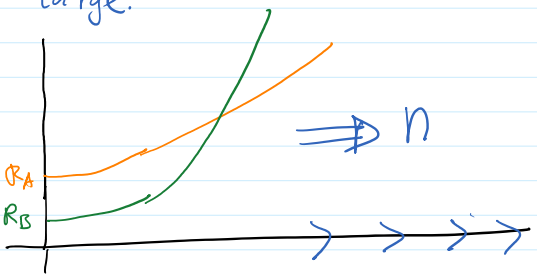
R: input: size of the program input
output: number of "operations" performed by the program.

operations: + − * /
[] = ==

$R_A(n)$ $R_B(n)$ n: size of the input.

- But!: • Which values of n to use?

no specific values.
we want to know what happens as 'n' becomes really large.



Graphs are decent fall

- we are going to ignore constant factors.

$R_{Bob}(n) = 3n^2$ $R_{keenan}(n) = 9n^2$

$R_{Bob} = R_{keenan}$ for our purposes.

we say that we care about the "rate-of-growth" "order-of-growth" of the function.
not their values at a specific input size n

$R_A(n)$: n is the size of the input.

$R_A(n)$: n is the size of the input.

Which input?

- ★ worst input: assume the input that takes the most operations.
- best input:
- average cost:
- amortized cost: same operation sometimes is expensive, sometimes is cheap.

E.G. expanding an array.

- ASYMPTOTIC NOTATION

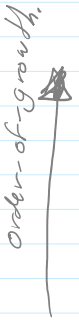
Informal definitions

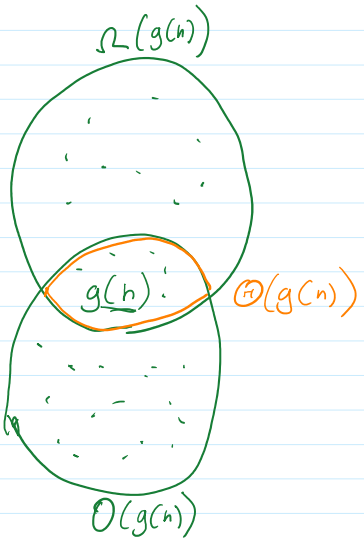
If $g(n)$ is a function

$O(g(n))$ is the set of all functions with lower or the same order-of-growth.

$\Omega(g(n))$ is the set of all functions with greater or the same order-of-growth.

$\Theta(g(n))$ is the set of all function with the same order-of-growth.





e.g. functions in $O(n^2)$

- $n \in O(n^2)$
- $3n^2 \in O(n^2)$
- $100n^2 + 27n + 3 \in O(n^2)$
- $n^3 \in \Omega(n^2)$
- $3n^2 \in \Omega(n^2)$
- $n! \in \Omega(n^2)$
- $3n^2 \in \Theta(n^2)$

DEF $O(g(n))$

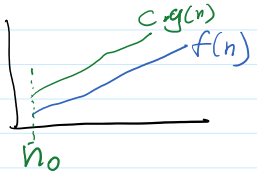
- We say that $f(n) \in O(g(n))$ if

there exists constants C and n_0 such that for every $n > n_0$

$$f(n) \leq C \cdot g(n)$$

- C allows us to ignore constant factors
- there is little restriction on where to get C and n_0

In English: $f(n)$ is bounded from above by a multiple of $g(n)$



E.G. $n^2 \in O(3n^2 + n)$

Proof: $C = 1$, $n_0 = 1$

$n^2 \leq 3n^2 + n$ ✓

E.G. $3n^2 + n \in O(n^2)$

Proof = $n_0 = 1$, $C = 4$

$3n^2 + n \leq 4 \cdot (n^2)$

$3n^2 + n \leq 4n^2$

$3n^2 + n \leq 3n^2 + 1n^2$

$n \leq n^2$ ✓

DEF: we say that $f(n) \in \Omega(g(n))$ if there exists constants C and n_0 such that for every $n > n_0$

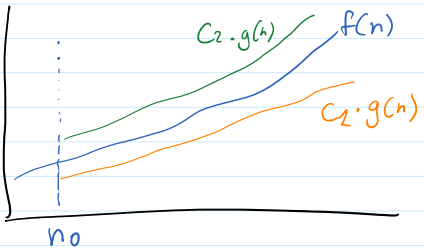
$$f(n) \geq C \cdot (g(n))$$

In English: $f(n)$ is bounded from below by a multiple of $g(n)$

DEF: we say that $f(n)$ is $\Theta(g(n))$ if there exists constants C_1, C_2 and n_0 such that for every $n > n_0$

$$C_1 \cdot g(n) \leq f(n) \leq C_2 \cdot g(n)$$

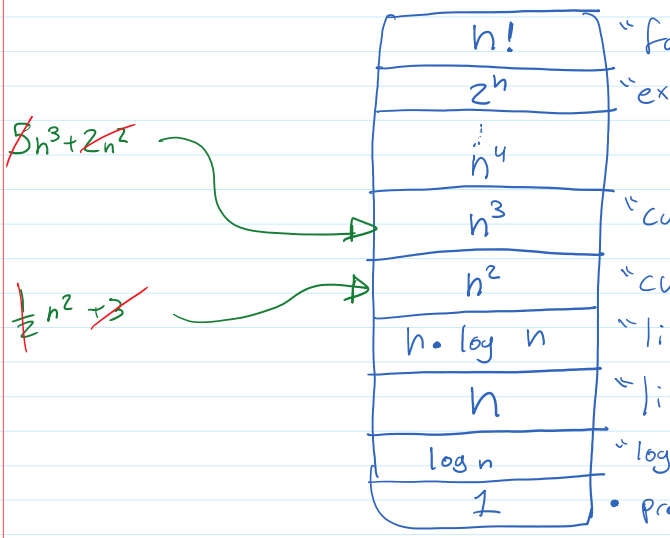
English: $f(n)$ is bounded from above and below by multiples of $g(n)$



THE BASIC EFFICIECY CLASSES:

Comparing functions

$R_{kevin}(n) = 5n^3 + 2n^2$ $\quad$ $R_{mario}(n) = \frac{1}{2}n^2 + 3$



- "factorial" - generate all permutations of a collection
- "exponential" - generate subsets of a collection.
- "cubic" - 3 nested loops.
- "quadratic" - 2 nested loops.
- "linearithmic" - divide and conquer algorithms.
- "linear" do one thing for every element of input
- "logarithmic". split input and only look at one part.
- program without loops.

MORE MATH : USEFUL PROPERTIES OF ORDER-OF-GROWTH

what if you need to run kevin followed by Mario.

$R_{kevin}(n) = 5n^3 + 2n^2$ $\quad$ $R_{mario}(n) = \frac{1}{2}n^2 + 3$

$R_{KM}(n) = R_{kevin}(n) + R_{mario}(n)$

- if you have function $f_1(n)$, $f_2(n)$, $g_1(n)$, $g_2(n)$ and you know $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$ then:

$$f_1(n) + f_2(n) \in O(\max\{g_1(n), g_2(n)\})$$

therefor

$$R_{KM}(n) = R_{kevin}(n) + R_{mario}(n) \in O(\max\{n^3, n^2\})$$

$$R_{KM}(n) \in O(n^3)$$

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

- Limits and the order-of-growth.

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \begin{cases} 0 & \text{then } f(n) \text{ has a smaller order-of-growth than } g(n) \\ c & \text{then } f(n) \text{ has the same order-of-growth than } g(n) \\ \infty & \text{then } f(n) \text{ has a larger order-of-growth than } g(n) \end{cases}$$

* it does not exists

- Allows us to use convenient results from calculus.
  - L'Hôpital rule:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$$

  - Stirling's formula

$$n! \approx \sqrt{2\pi \cdot n}\left(\frac{n}{e}\right)^n \quad \text{for large values of } n.$$

— EOF —