Discrete Math

Lecture 36

Big O, Big Ω and Big Θ

why?

we want to estimate the amount / the # operations an algorithm might use in order to run.

"complexity" or "run-time" of an alg. will be a function n

How does this grow as $n \to \infty$??

Big-O, Big-Ω, Big-Θ notation gives us a way to answer this!

Big-O notation

$f(x)$ is $O(g(x))$ means

$f(x)$ is eventually smaller than a const. $\cdot\, g(x)$

i.e. "f is $O(g)$" means $\exists\, k, C \in \mathbb{R}^+$,

$$\forall x > k, \quad |f(x)| \leq C \cdot |g(x)|$$

sometimes called "witnesses"

ex] $f(x) = x^2 + 2x + 1$

claim: $f(x)$ is $O(x^2)$

we need to find a k, C

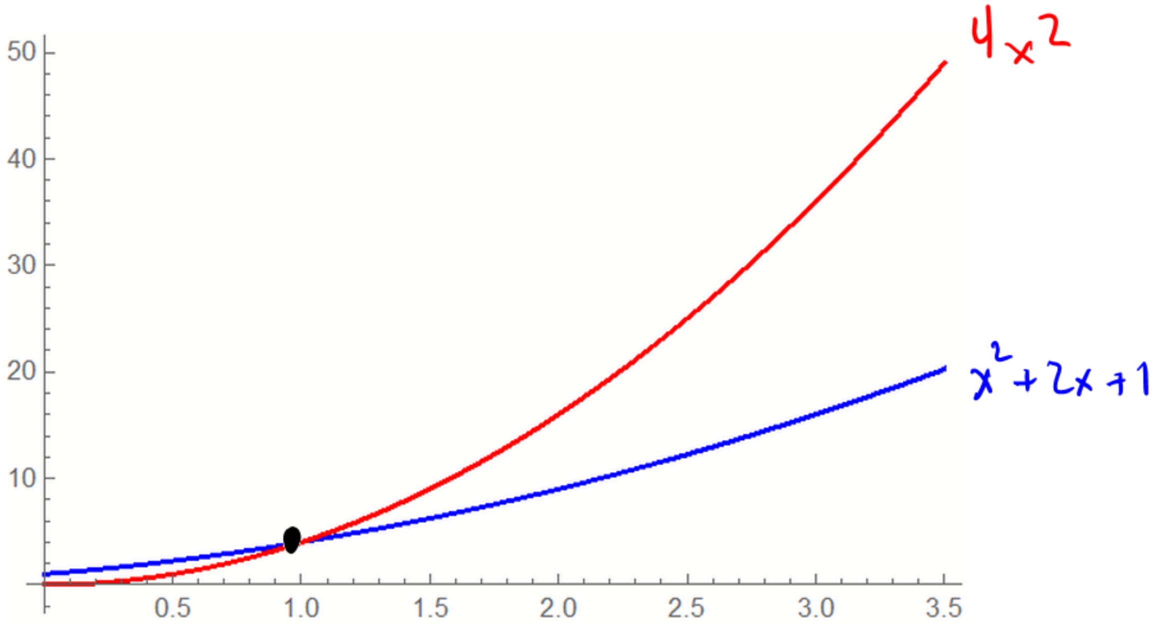
check: $C = 4$, $k = 1$

$$\forall x > 1, \quad |x^2 + 2x + 1| \leq 4x^2$$

since $x > 1$, $x^2 > x \rightarrow x < x^2$

$$x^2 + 2x + 1 \le x^2 + 2x^2 + x^2$$

$$= 4x^2$$



$$\underbrace{x^2 + 2x + 1 \le 4 \cdot x^2 \qquad \forall\, x > 1}$$

$$x^2 + 2x + 1 \text{ is } O(x^2)$$

We can generalize the above example to any polynomial:

ex) $7x^8 - 33x^6 + x^5 - 3x^4 + x^2 + \pi$ is $O(x^8)$

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \text{ is } O(x^n)$$

ex| What's a useful big-O estimate for the sum of the first n natural numbers?

$n=1$: 1

$n=2$: $1+2 = 3$

$n=3$: $1+2+3 = 6$

$\vdots$

formula: $\displaystyle\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$ is $O(n^2)$

another way: $\displaystyle\sum_{i=1}^{n} i = 1+2+3+\cdots+n \leq n+n+n+\cdots+n$

$= n^2$

$\displaystyle\sum_{i=1}^{n} i$ is $O(n^2)$

ex) $n!$ is $O(\quad)$

$$n! = n \cdot (n-1)(n-2)(n-3)\cdots 2 \cdot 1$$

$$\leq n \cdot n \cdot n \cdot n \cdots n \cdot n$$

$$= n^n$$

$C = 1$ $\quad$ $k = 1$

$$\underbrace{\forall\, n > 1, \quad n! < 1 \cdot n^n}$$

$n!$ is $O(n^n)$

extra note $\quad n! \leq n^n$

apply ln to both sides: $\underbrace{\ln(n!) \leq n \cdot \ln(n)}$

$\ln(n!)$ is $O(n \cdot \ln(n))$

A little more about Big-O

- $\sum_{i=1}^{n} i$ is $O(n^2)$ $\Rightarrow$ $\sum_{i=1}^{n} i$ is $O(n^3)$

Big-O estimates aren't unique or "tight"

- f_1 is $O(g_1)$ $\wedge$ f_2 is $O(g_2)$

then $(f_1 + f_2)$ is $O\left(\max\{g_1, g_2\}\right)$

in particular, if f_1 & f_2 are $O(g)$

then $f_1 + f_2$ is $O(g)$

ex) $f_1(n) = \sum_{i=1}^{n} i$ $\qquad f_2(n) = 5n^2 + 3n + 1$

$\left(f_1(n) + f_2(n)\right)$ is $O(n^2)$

- if f_1 is $O(g_1)$ and f_2 is $O(g_2)$

then $f_1 \cdot f_2$ is $O(g_1 \cdot g_2)$

Big-Omega Big-Ω

$f(x)$ is $\Omega(g(x))$ means

$\exists$ k, C, if $x > k$ then

$$|f(x)| \geq C \cdot |g(x)|$$

Fact f is $\Omega(g)$ $\Leftrightarrow$ g is $O(f)$

Big-Theta Θ

f is $\Theta(g)$ means two things

- f is $O(g)$
- f is $\Omega(g)$

i.e. f is $\Theta(g)$ means the graph of f is eventually sandwiched between the graph of $c_1 \cdot g$ & $c_2 \cdot g$