1 growth rates

we learned about algorithms, however, we don't really know how to compare them – which one is more efficiency, or fast, etc. in this section we introduce the big-o notation, which is used to measure the growth rate of an algorithm.

1.1 big-o notation

roughly speaking, the big-o notation describes how fast a function $f(x)$ grows – and the growth of this function will never exceed the growth of another function $g(x)$.

1.1.1 definition

let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. we say that $f(x)$ is $O(g(x))$ if there are constants C and k such that $|f(x)| \leq C|g(x)|$ whenever $x > k$. we read this as: "$f(x)$ is big-o of $g(x)$".

in simpler words, the definition of the growth rate of $f(x)$ is $O(g(x))$ means that $f(x)$ grows slower than some fixed multiple(C) of $g(x)$ as x grows without bound past k.

we call the constants C and k the witnesses – and there are infinite witnesses that exist when $f(x)$ is $O(g(x))$. if we find a pair of witnesses (C, k) that work, then any C' such that $C' > C$ will also work; when we use the same k. likewise, any k' such that $k' > k$ will work, when using the same C.

1.1.2 growth rate of polynomials

for polynomial functions, we can drop multiplicative constants and lower-order terms. in short, a $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$ is $O(x^n)$.

we can prove this by using the triangle inequality ($|a + b| \leq |a| + |b|$). if $x > 1$ we have

$$\begin{aligned} |f(x)| &= |a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0\ | \\ &\leq |a_n| x^n + |a_{n-1}| x^{n-1} + ... + |a_1| x + |a_0| \\ &= x^n \left(|a_n| + \frac{|a_{n-1}|}{x} + ... + \frac{|a_1|}{x^{n-1}} + \frac{|a_0|}{x^n} \right) \\ &\leq x^n (|a_n| + |a_{n-1}| + ... + |a_1| + |a_0|) \end{aligned}$$

this shows that $|f(x)| \leq C x^n$, where $C = |a_n| + |a_{n-1}| + ... + |a_0|$ whenever $x > 1$. hence, the witnesses $C = |a_n| + |a_{n-1}| + ... + |a_0|$ and $k = 1$ show that $f(x)$ is $O(x^n)$.

1.1.3 growth rate of combined functions

for the sum of functions, suppose that $f_1(x)$ is $O(g_1(x))$ and that $f_2(x)$ is $O(g_2(x))$. then, $(f_1 + f_2)(x)$ is $O(g(x))$, where $g(x) = \max(|g_1(x)|, |g_2(x)|)$ for all x.

for products, it is similar: if $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1 f_2)(x)$ is $O(g_1(x)g_2(x))$. in plain english, the *product* of two functions has a growth rate equal to the *product* of their individual growth rates.

1.1.4 finding the witnesses

finding the constants C and k is usually done by simplifying the function $f(x)$ so it looks like $g(x)$. the main idea is to "round up":

1. choose the k: we usually choose a simple value like $k = 1$ to make sure x is positive.
2. "round up" the terms: replace the lower-order terms (like x or constants) with the highest-order term (like x^2)
   - because for $x > 1$, $x^n \leq x^{n+1}$.
3. add them: sum the coefficients to find C.

1.1.4.1 example

show that $f(x) = 3x^2 + 8x + 4$ is $O(x^2)$.

1. let $k = 1$. this means that for $x > 1$, $|f(x)| \leq C|g(x)|$.
2. since $x > 1$, we know that $x < x^2$ and $1 < x^2$, therefore, we can replace x in the inequality with x^2, and it will still be true:
   - $|3x^2 + 8x + 4| \leq |3x^2 + 8x^2 + 4x^2|$
3. we rewrite the inequality (by triangle inequality):
   - $|3x^2 + 8x + 4| \leq |3x^2| + |8x^2| + |4x^2|$
4. combine the terms on the right side (and since we know that $x > 1$, we can drop the absolute value):
   - $|3x^2 + 8x + 4| \leq 15x^2$
5. with $g(x) = x^2$, on the right side, we can see that $C = 15$. ■

therefore, we have found that $C = 15$ and $k = 1$.

1.1.5 is it the smallest bound?

sometimes, we need to prove that a $f(x)$ is not $O(g(x))$; which means that $f(x)$ grows faster than $g(x)$, and no matter how big of C you choose, $f(x)$ will eventually grow faster than $g(x)$.

to prove $f(x)$ is not $O(g(x))$, we must show that for *every* pair of witnesses C and k, there exists some $x > k$ such that: $|f(x)| > C|g(x)|$.

we can usually prove this by contradiction:

1. assume that $f(x)$ is $O(g(x))$.
2. previous step implies that $|f(x)| \leq C|g(x)|$ for some constant C.

3. simplify the inequality to show that it is impossible (usually $x \leq C$).
4. since x can grow infinitely large, it can't be limited by a constant C. contradiction.

1.1.5.1 example

show that x^2 is not $O(x)$.

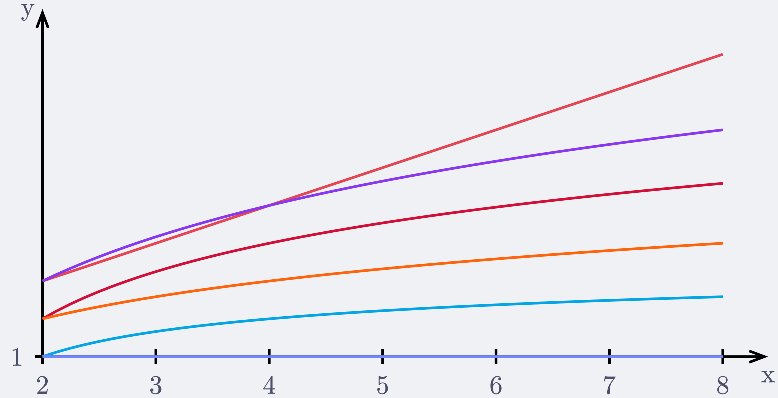
1. assume x^2 is $O(x)$.
2. therefore, there exists witnesses C and k such that $|x^2| \leq C|x|$ for all of $x > k$.
3. let $k = 1$.
4. the inequality then becomes $x^2 \leq Cx$, dropping the inequality.
5. divide both sides by x, the inequality becomes $x \leq C$.
6. since x grows to infinity, it cannot be smaller than a constant ($x = C + 1$).

therefore, x^2 is not $O(x)$ by contradiction. ■

1.1.6 common $g(x)$'s

here are some common growth rates, ordered from the fastest growth to slowest. note that $O(n!)$ isn't included in the graph because of a technical issue, sorry!

- $O(n!)$
- $O(2^n)$
- $O(n^2)$
- $O(n \log n)$
- $O(n)$
- $O(\log n)$
- $O(1)$



1.2 other growth notations

there are other notations that are closely related to the big-o notation. we introduce the big-omega notation, which is the opposite of big-o; and the big-theta notation.

1.2.1 big-omega notation

sometimes, we want to know if the function grows at least as fast as $g(x)$ (a lower bound). in computer science, we use big-o as a measurement of the "worst case", and big-omega as a measurement of "best case".

1.2.1.1 definition

let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. we say that $f(x)$ is $\Omega(g(x))$ if there are constants C and k with C *positive* such that $f(x) \geq C|g(x)|$ whenever $x > k$. we read this as: "$f(x)$ is big-omega of $g(x)$".

1.2.1.2 relation to big-o

there is a connection between big-o and big-omega. specifically, $f(x)$ is $\Omega(g(x))$ *if and only if* $g(x)$ is $O(f(x))$. to put it simply, if f grows *faster* than g (f is $\Omega(g)$), then g grows *slower* than f (g is $O(f)$).

1.2.2 big-theta notation

if $O(g(x))$ describes the *upper bound* of $f(x)$ and $\Omega(g(x))$ describes the *lower* bound of $f(x)$, then $\Theta(g(x))$ describes the *exact* growth of $f(x)$.

1.2.2.1 definition

let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. we say that $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$. when $f(x)$ is $\Theta(g(x))$, we say that f is big-theta of $g(x)$, that $f(x)$ is of *order* $g(x)$, and that $f(x)$ and $g(x)$ are of the *same order*. instead of two constants C and k, we have three constants C_1, C_2, and k such that: $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$, whenever $x > k$.