Lecture 15

Chapter 3. The Fundamentals

3.2 The Growth of Functions

Efficiency of Algorithms

- Intuitively we see that binary search is much faster than linear search, but how do we analyze the efficiency of algorithms formally?

- Use methods of ***algorithmic complexity***, which utilize the order-of-growth concepts

University of Hawaii

3.2 Growth of Functions

- Analysis of an algorithm
  - Derive estimates for the time and space needed to execute the algorithm.
- Complexity of an algorithm
  - Amounts of time and space required to execute the algorithm
    → function of the input: difficult to obtain an explicit formula
    → instead of dealing with the input, function of the size of the input
- What really matters in comparing the complexity of algorithms?
  - We only care about the behavior for **large** problems.
  - Even bad algorithms can be used to solve small problems.
  - Ignore implementation details such as loop counter increment, etc.

Growth of Functions (cont.)

- <u>**Goal**</u>: To introduce the **big-*O* notation** and to show how to estimate the growth of functions using this notation and thereby to estimate the complexity (and hence the running time) of algorithms.

- For functions over numbers, we often need to know a rough measure of ***how fast a function grows***.
- If $f(x)$ is *faster growing* than $g(x)$, then $f(x)$ always eventually becomes larger than $g(x)$ *in the limit* (for <u>large enough</u> values of x).
- Useful in engineering for showing that one design *scales* better or worse than another.

University of Hawaii

Orders of Growth: Motivation

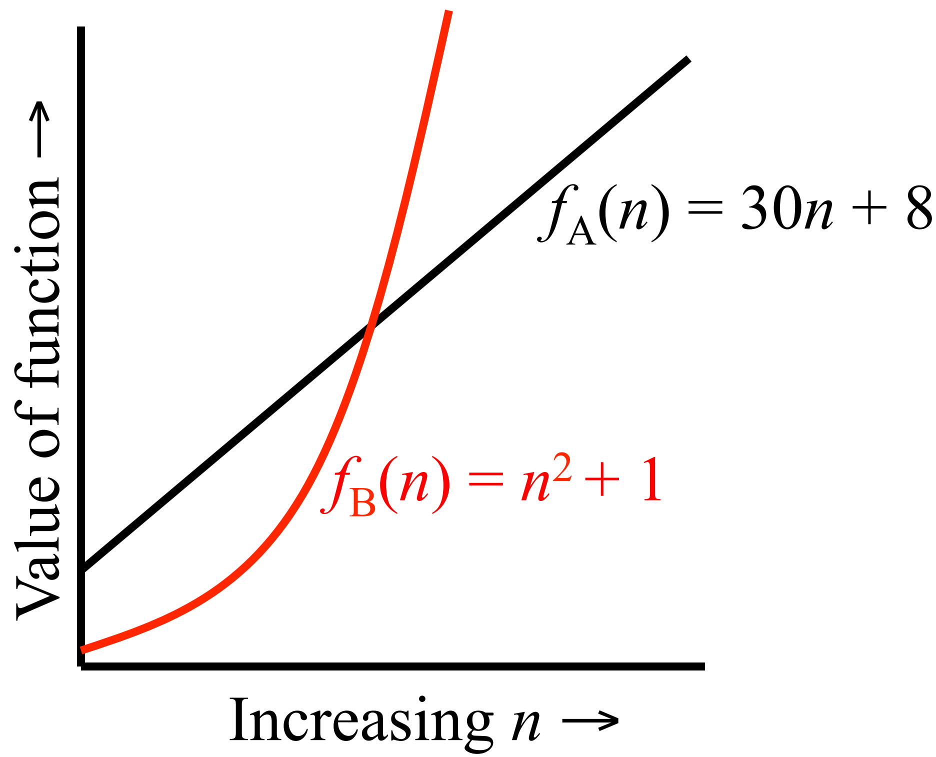
- Suppose you are designing a web site to process user data (*e.g.*, financial records).
- Suppose database program A takes $f_A(n) = 30n + 8$ microseconds to process any n records, while program B takes $f_B(n) = n^2 + 1$ microseconds to process the n records.
- Which program do you choose, knowing you'll want to support millions of users?

A

University of Hawaii

Visualizing Orders of Growth

- On a graph, as you go to the right, the faster growing function always eventually becomes the larger one...



Concept of Order of Growth

- We say $f_A(n) = 30n + 8$ is (*at most*) *order of n*, or $O(n)$.
  - It is, at most, roughly *proportional* to n.
- $f_B(n) = n^2 + 1$ is *order of* n^2, or $O(n^2)$.
  - It is (at most) roughly *proportional* to n^2.
- Any function whose *exact* (tightest) order is $O(n^2)$ is faster-growing than any $O(n)$ function.
  - Later we will introduce Θ for expressing *exact* order.
- For large numbers of user records, the order n^2 function will always take more time.

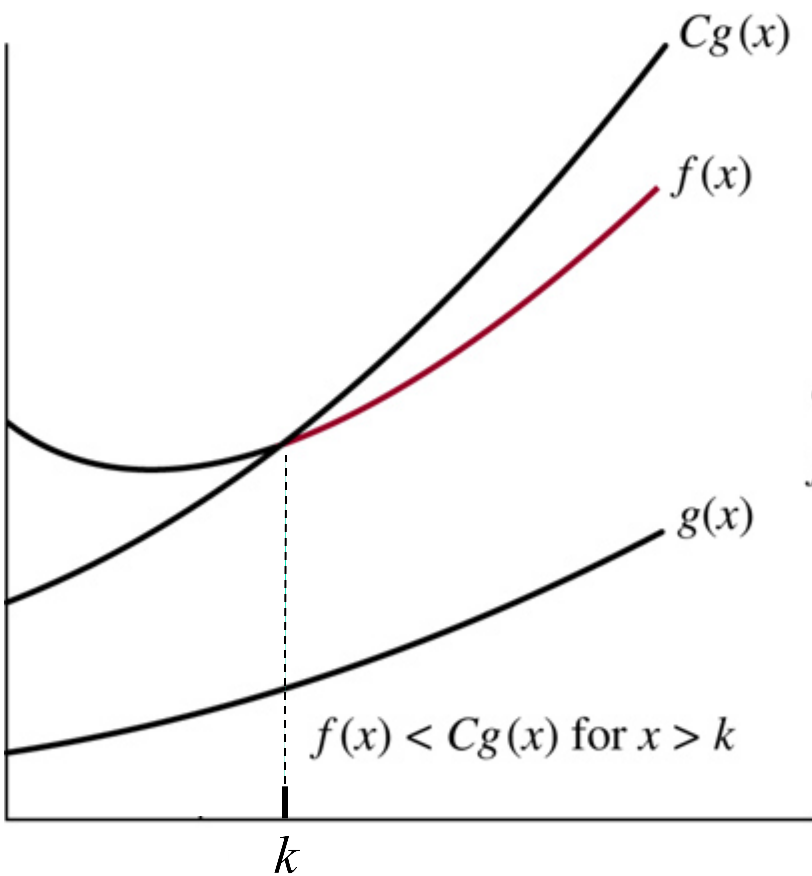
Big-*O* Notation

- Let *f* and *g* be functions **R** (or **Z**) → **R**.
- Definition: "***f* is big-*O* of *g***" (or "*f* is **in the class** *O*(*g*)") if

$$\exists C,k \text{ such that } |f(x)| \le C|g(x)|\ \forall x>k.$$

  - "Beyond some point *k*, function *f* is at most a constant *C* times *g* (*i.e.*, proportional to *g*)." : *f* is bounded from above by *g*
- "*f* is *at most order g*", or "*f* is *O*(*g*) (*f* is big-oh of *g*)", or "*f* = *O*(*g*)" all just mean that $f \in O(g)$.
- Often the phrase "at most" is omitted.
- The constants *C* and *k* are called ***witnesses*** to the relationship *f*(*x*) is *O*(*g*(*x*)).

Big-*O* Notation Illustration



The part of the graph of $f(x)$ that satisfies $f(x) < Cg(x)$ is shown in color.

Points about the Definition

University of Hawaii

- Note that *f* is *O*(*g*) so long as *any* values of *C* and *k* exist that satisfy the definition.
- But: The particular *C*, *k*, values that make the statement true are *not* unique:
  Any larger value of *C* and/or *k* will also work.
- You are **not** required to find the smallest *C* and *k* values that work. (Indeed, in some cases, there may be no smallest values!)

However, you should **prove** that the values you choose do work.

"Big-O" Proof Example I

- Show that $30n + 8$ is $O(n)$.
  - Show $\exists C,k$ such that $\forall n > k$, $30n + 8 \le Cn$.
    - Let $k = 8$. Assume $n > 8$ $(= k)$.

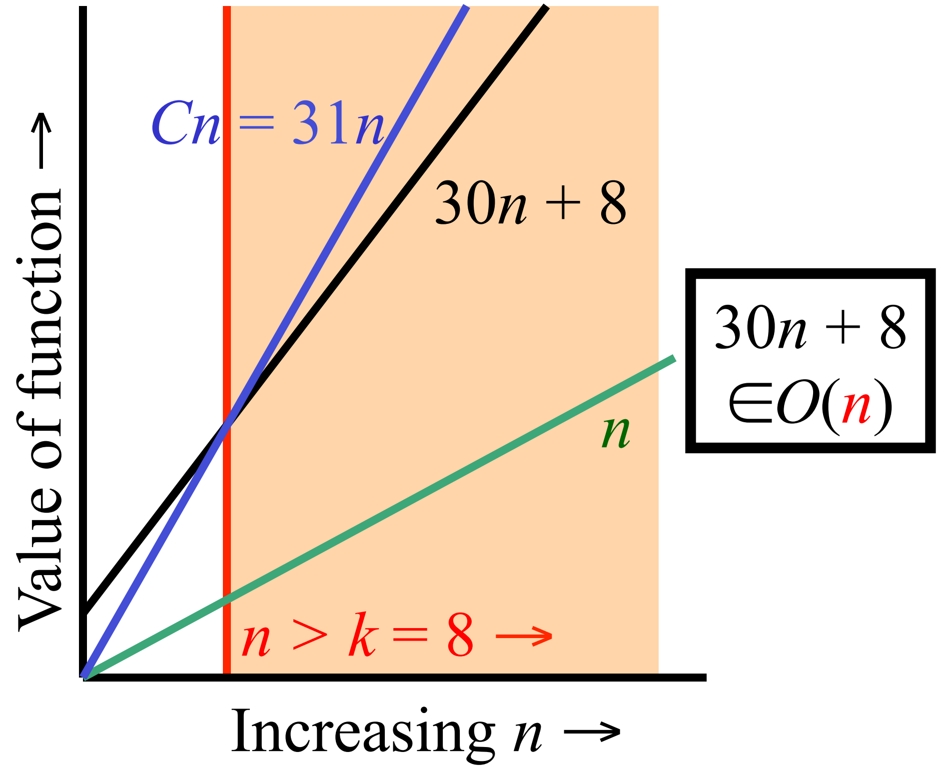
      Then, $30n + 8 < 30n + n = 31n$.

      Therefore, we can take $C = 31$ and $k = 8$ to show that $30n + 8$ is $O(n)$.

Big-*O* Example, Graphically

University of Hawaii

- Note $30n + 8$ isn't less than n *anywhere* ($n > 0$).
- It isn't even less than $31n$ *everywhere*.
- But it *is* less than $31n$ <u>everywhere to the right of $n = 8$</u>.



"Big-O" Proof Examples II

- Show that $n^2 + 1$ is $O(n^2)$.
  - Show $\exists C,k$ such that $\forall n > k$, $n^2 + 1 \le Cn^2$.
    - Let $k = 1$. Assume $n > 1$ $(= k)$.

      Then, $n^2 + 1 < n^2 + n^2 = 2n^2$.

      Take $C = 2$ and $k = 1$, then
      $\forall n > k$, $n^2 + 1 \le Cn^2$ (i.e. $n^2 + 1$ is $O(n^2)$).

University of Hawaii

Big-*O* and Polynomials

- **Theorem 1**: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ where $a_0, a_1, \ldots, a_{n-1}, a_n$ are real numbers (i.e. $f(x)$ is a polynomial of degree n). Then $f(x)$ is $O(x^n)$.
  - ***Proof***: Using the triangular inequality ($|a + b| \leq |a| + |b|$), if $x > 1$ we have

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

  - This shows that $|f(x)| \leq Cx^n$,
    where $C = |a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|$ whenever $x > 1$.
  - Hence, $f(x)$ is $O(x^n)$.

Examples

- $1 + 2 + \cdots + n \le \underbrace{n + n + \cdots + n + n}_{n \text{ terms}} = n^2$.
  It follows that $1 + 2 + \cdots + n$ is $O(n^2)$, taking $C = 1$ and $k = 1$ as witnesses.

Note: $$1 + 2 + \cdots + n = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$
$$= \frac{1}{2}n^2 + \frac{1}{2}n$$

More Big-*O* Examples

- $n! = 1 \cdot 2 \cdot 3 \cdots n \le n \cdot n \cdot n \cdots n = n^n$

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

- $\log n! \le \log n^n = n \log n$

  $\Rightarrow \log n!$ is $O(n \log n)$

- $n < 2^n$ whenever n is a positive integer

  $\Rightarrow n$ is $O(2^n)$ (n is also $O(n)$)

- $\log n < n$

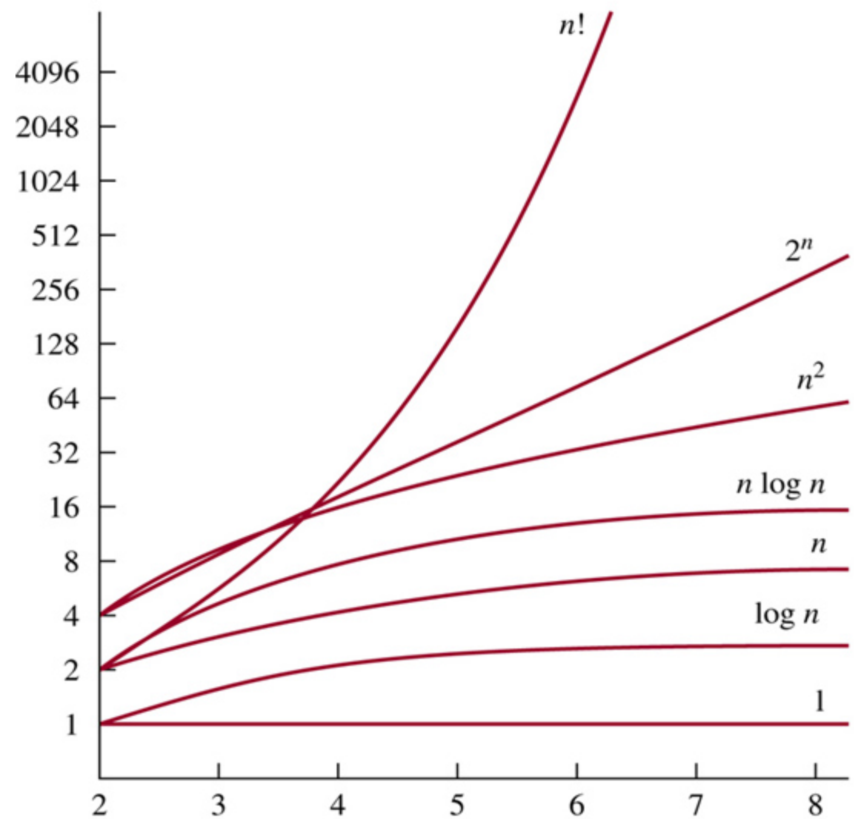
  $\Rightarrow \log n$ is $O(n)$ ($\log n$ is also $O(\log n)$)

University of Hawaii

Order of Growth of Functions

- Important complexity classes

$$
\begin{aligned}
O(1) &\subseteq O(\log n) \\
&\subseteq O(n) \\
&\subseteq O(n\log n) \\
&\subseteq O(n^2) \\
&\subseteq O(c^n) \\
&\subseteq O(n!)
\end{aligned}
$$



A Display of the Growth of Functions Commonly Used in Big-O Estimates

Useful Facts about Big-*O*

- Big *O*, as a relation, is transitive:
  $f \in O(g) \wedge g \in O(h) \rightarrow f \in O(h)$
- $\forall c > 0$, $O(cg) = O(g + c) = O(g - \mathrm{c}) = O(g)$
  (c is a positive constant)
- $f_1 \in O(g_1) \wedge f_2 \in O(g_2) \rightarrow$
  - $f_1 f_2 \in O(g_1 g_2)$
  - $f_1 + f_2 \in O(g_1 + g_2)$
    $= O(\max(g_1, g_2))$
    $= O(g)$ where $g = \max(|g_1|, |g_2|)$
- If $f \in \mathrm{O}(h)$ and $g \in \mathrm{O}(h)$, then $f + g \in \mathrm{O}(h)$.

An Example

- **Example 9**: Give a big-O estimate for $f(x) = (x + 1)\log(x^2 + 1) + 3x^2$.
  - $x + 1$ is $O(x)$
  - $\log(x^2 + 1)$ is $O(\log x)$
    - $\log(x^2 + 1) \leq \log(2x^2) = \log 2 + \log x^2$

when $x > 1$

$= \log 2 + 2\cdot\log x \leq 3\cdot\log x$, if $x > 2$

  - Therefore, $(x + 1)\log(x^2 + 1)$ is $O(x \log x)$
  - $3x^2$ is $O(x^2)$
  - $f(x) = (x+1)\log(x^2+1) + 3x^2$ is $O(\max(x\cdot\log x, x^2))$. Because $x\cdot\log x \leq x^2$, for $x > 1$, it follows that $f(x)$ is $O(x^2)$.

Big-Ω Notation

- Let *f* and *g* be functions from the set of integers or the set of real numbers to the set of real numbers.

- $f(x)$ is $\Omega(g(x))$ if there are positive constants C and k such that $|f(x)| \geq C|g(x)|$ whenever $x > k$.

- This is read as "$f(x)$ is big-Omega of $g(x)$."

- $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$

Big-Θ Notation

- Let *f* and *g* be functions from the set of integers or the set of real numbers to the set of real numbers.
- If $f \in O(g)$ and $f \in \Omega(g)$, then we write $f \in \Theta(g)$ and say "*f* is big-Theta of *g*" and also "*f is (exactly) of order g*".
- Another, equivalent definition:
  $\Theta(g) \equiv \{f : \mathbf{R}\ (\text{or}\ \mathbf{Z}) \rightarrow \mathbf{R} \mid$
  $\exists C_1, C_2, k > 0\ \forall x > k\colon C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|\ \}$
  - "Everywhere beyond some point *k*, *f*(*x*) lies in between two multiples of *g*(*x*)."

Big-Θ and Polynomial

- **Theorem 4**
  - Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$,
    where $a_0, a_1, \ldots, a_n$ are real numbers with $a_n \neq 0$.
  - Then $f(x)$ is of order x^n.

- **Example:** $f(n) = 60n^2 + 5n + 1$
  - $60n^2 + 5n + 1 \leq 60n^2 + 5n^2 + n^2 = 66n^2$ for $n > 1$
    $\therefore f(n) = O(n^2)$
  - $60n^2 + 5n + 1 \geq 60n^2$ for $n > 1$
    $\therefore f(n) = \Omega(n^2)$
  - Therefore, $f(n) = \Theta(n^2)$

Θ Example

- Determine whether: $\left(\sum_{i=1}^{n} i\right) \overset{?}{\in} \Theta(n^2)$
- Quick solution:

$$\left(\sum_{i=1}^{n} i\right) = n(n+1)/2$$

$$= n\,\Theta(n)/2$$

$$= \Theta(n)\cdot\Theta(n)$$

$$= \Theta(n^2)$$

Θ Example: Alternative Proof

- $f(n) = 1 + 2 + \cdots + n$

  $1 + 2 + \cdots + n \leq n + n + \cdots + n = n^2$ for $n > 1$ $\rightarrow f(n) = O(n^2)$

  $1 + 2 + \cdots + n \geq 1 + 1 + \cdots + 1 = n$ $\rightarrow f(n) = \Omega(n)$

  - We cannot deduce a Θ-estimate for $1 + 2 + \cdots + n$, since the upper bound n^2 and the lower bound n are not equal. We must be craftier in deriving a lower bound.

$$1 + 2 + \cdots + n \geq \left\lceil \frac{n}{2} \right\rceil + \left(\left\lceil \frac{n}{2} \right\rceil + 1 \right) + \cdots + n$$

Ignore the first half of the terms

$$\therefore\ 1 + 2 + \cdots + n = \Omega(n^2)$$

$$\therefore\ 1 + 2 + \cdots + n = \Theta(n^2)$$

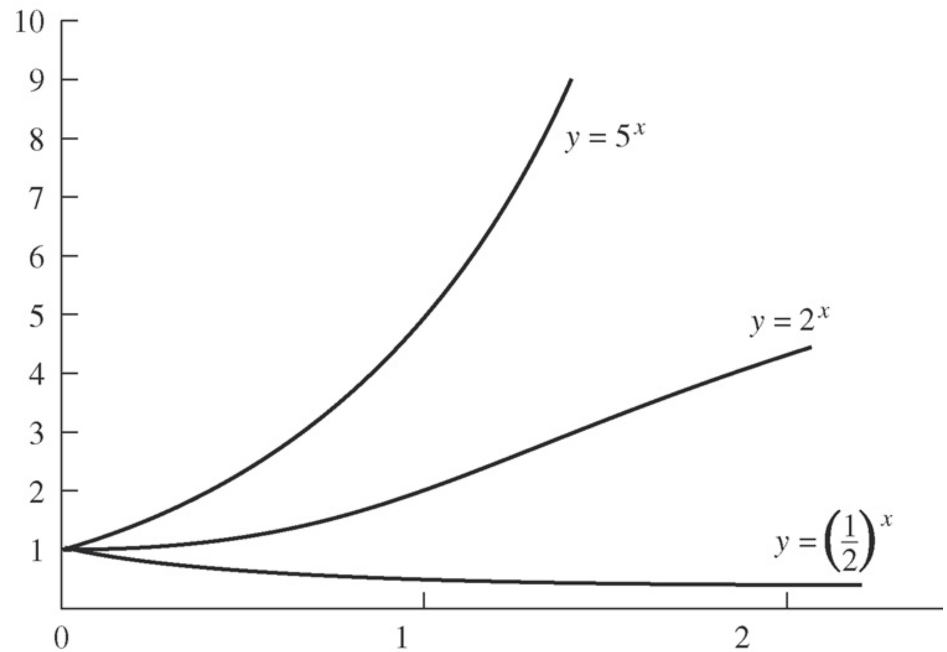
Exponential Functions

- **Exponential function to the base *b***

$$b^n = \underbrace{b \cdot b \cdot b \cdots b}_{n \text{ times}}$$

- **<u>Theorem</u>**
  - $b^{x+y} = b^x b^y$
  - $(b^x)^y = b^{xy}$

- **Read Appendix 2**



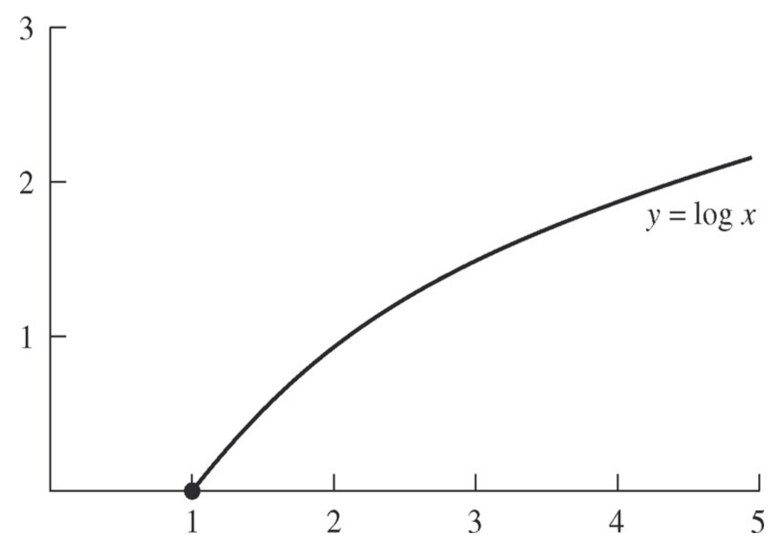
Graphs of the exponential functions to the bases ½, 2, and 5

Logarithmic Functions

University of Hawaii

- If $b > 1$ and $b \in \mathbf{R}$ then b^x is
  - Strictly increasing
  - One-to-one correspondence from **R** to nonnegative **R**
  - Inverse function of $y = b^x$: **Logarithmic function to the base *b***

$$y = \log_b x \quad (b^y = b^{\log_b x} = x)$$



The graphs of y = log *x*

- **Theorems**
  - $\log_b(xy) = \log_b x + \log_b y$
  - $\log_b(x^y) = y \log_b x$
  - Let $a,b \in \mathbf{R}$ greater than 1, and let $x \in \mathbf{R}^+$. Then, $\log_a x = (\log_b x)/(\log_b a)$