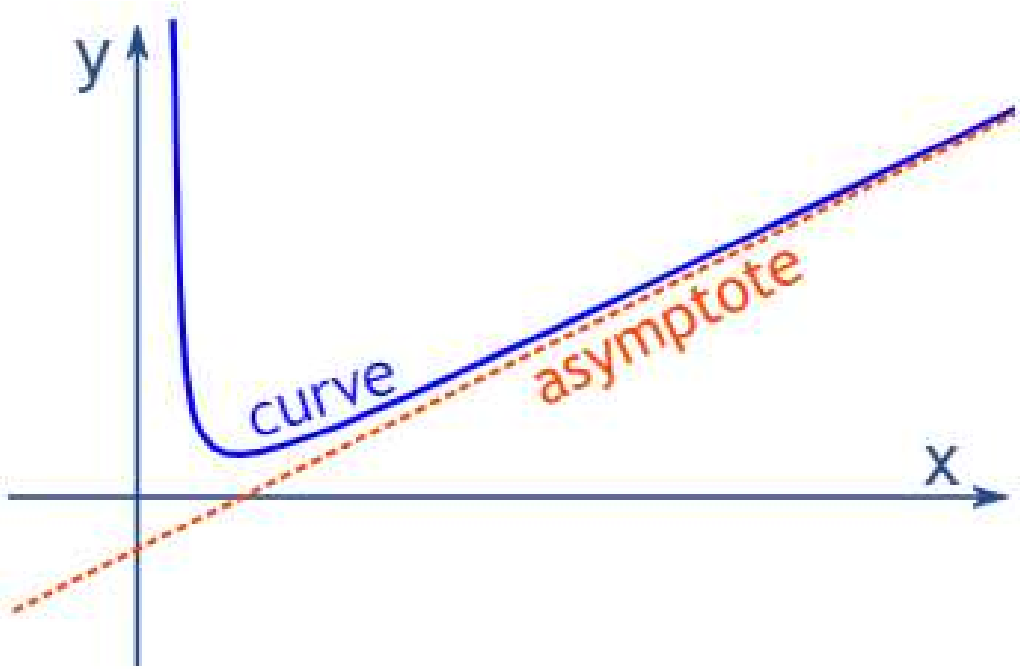
# Lecture 7

Time Complexity Measuring Notations: Growth Rate of Functions; Asymptotic Notations: Big O Notation, Sigma Notation, and Theta Notation

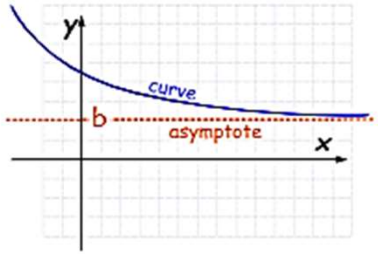
# Asymptote

› An asymptote is a **line** that a curve approaches, as it heads towards infinity:



# Asymptote Types

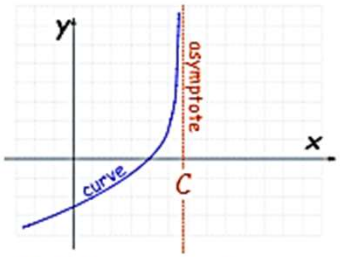
## Horizontal Asymptotes




It is a Horizontal Asymptote when:

as x goes to infinity (or −infinity) the curve approaches some constant value **b**

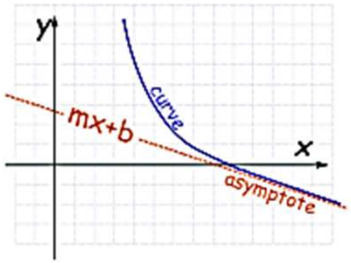
## Vertical Asymptotes




It is a Vertical Asymptote when:

as x approaches some constant value **c** (from the left or right) then the curve goes towards infinity (or −infinity).

## Oblique Asymptotes




It is an Oblique Asymptote when:

as x goes to infinity (or −infinity) then the curve goes towards a line **y=mx+b**

(note: m is not zero as that is a Horizontal Asymptote).

# Meanings of Asymptotic

Informally, the term asymptotic means approaching a value or curve arbitrarily closely (i.e., as some sort of limit is taken). A line or curve $A$ that is asymptotic to given curve $C$ is called the asymptote of $C$.

More formally, let $x$ be a continuous variable tending to some limit. Then a real function $f(x)$ and positive function $\phi(x)$ are said to be asymptotically equivalent, written $f \sim \phi$, if $\boxed{f/\phi \to 1}$ (1)

as the limit is taken.

Equivalently, consider the little-o asymptotic notation $o(x)$ that is one of the Landau symbols. Then $f = o(\phi)$ means that

$$\boxed{f/\phi \to 0} \quad (2)$$

as a limit is taken. The statement $f \sim \phi$ is then equivalent to

$$\boxed{f = \phi + o(\phi)} \quad (3)$$

or

$$\boxed{f = \phi(1 + o(1))} \quad (4)$$

These definitions can also be applied to the discrete case of $n$ an integer variable that tends to infinity, $f(n)$ a real function of $n$, and $\phi(n)$ a positive function of $n$.

# *Asymptotic Notation*

- ***O notation**: asymptotic "less than":*
  - *$f(n)=O(g(n))$ implies $f(n)$ "$\leq$" $g(n)$*
- ***$\Omega$ notation**: asymptotic "greater than":*
  - *$f(n)=\Omega(g(n))$ implies $f(n)$ "$\geq$" $g(n)$*
- ***$\Theta$ notation**: asymptotic "equality":*
  - *$f(n)=\Theta(g(n))$ implies $f(n)$ "$=$" $g(n)$*

# *Definition: O(g), at most order g*

*Let f, g are functions* $\mathbf{R} \rightarrow \mathbf{R}$.

- *We say that "f is at most order g", if:*

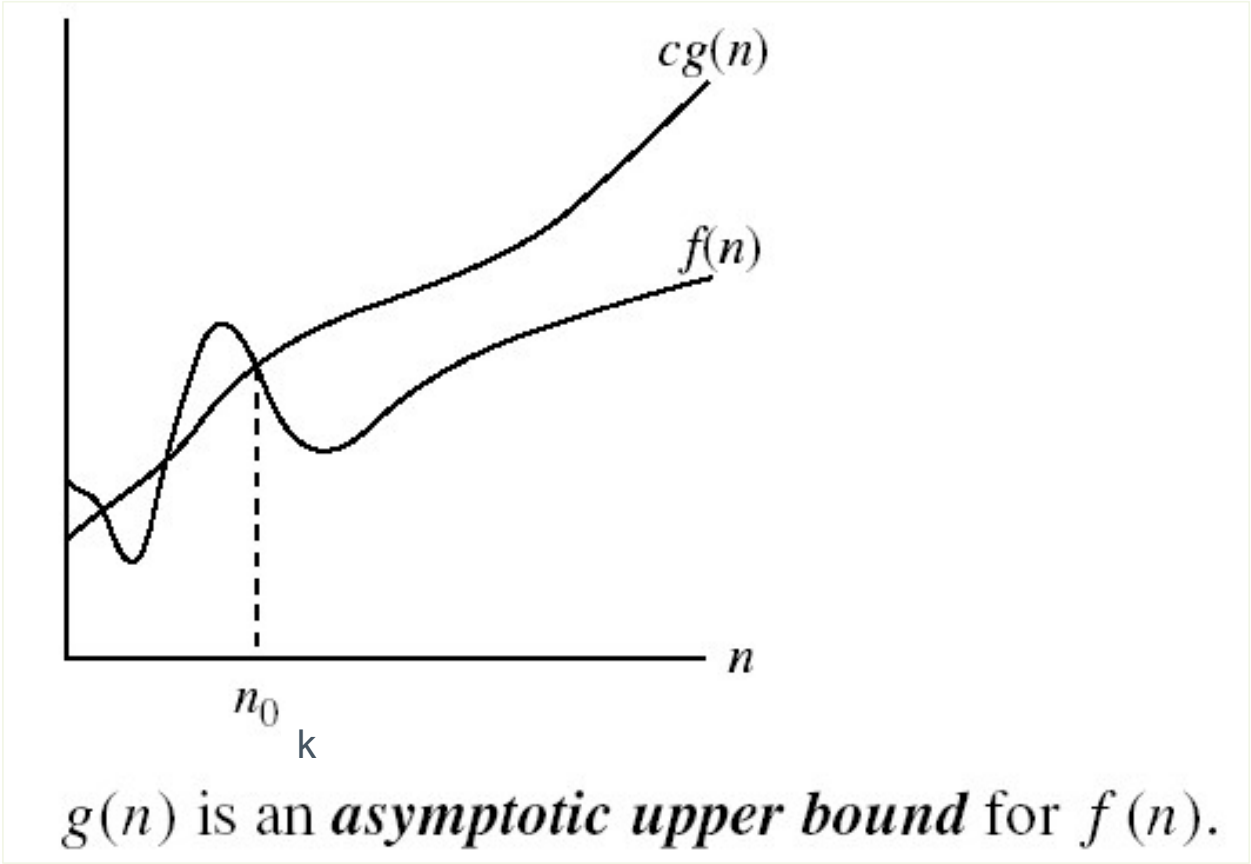
$$\exists\, c, k: f(x) \leq cg(x),\ \forall x > k$$

  - *"Beyond some point k, function f is at most a constant c times g (i.e., proportional to g)."*

- *"f is at most order g", or "f is O(g) ", or "f=O(g)" all just mean that* $f \in O(g)$.

- *Sometimes the phrase "at most" is omitted.*

# *Big-O Visualization*




$g(n)$ is an ***asymptotic upper bound*** for $f(n)$.

# *Points about the definition*

- *Note that f is O(g) as 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!)*

# "Big-O" Proof Examples

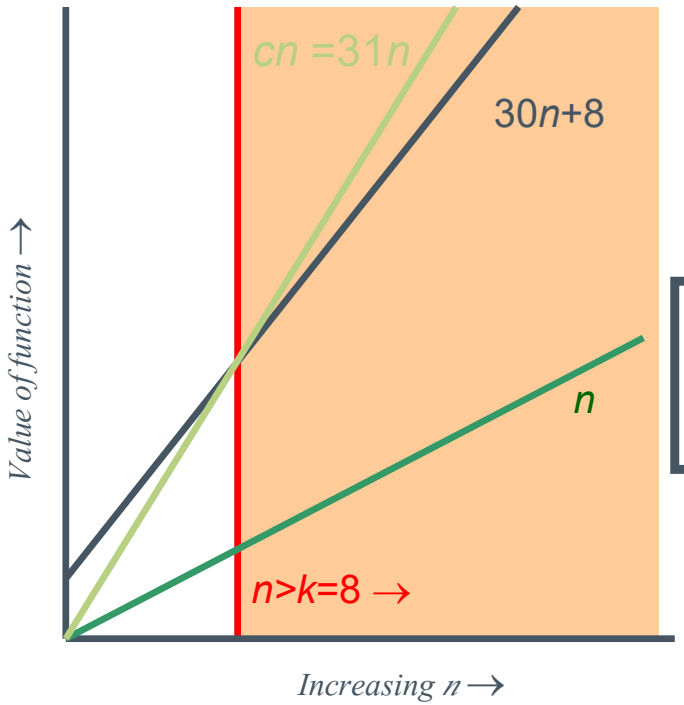
- *Show that $30n+8$ is $O(n)$.*
  - *Show $\exists c, k$: $30n+8 \leq cn$, $\forall n>k$.*
    - *Let $c=31$, $k=8$. Assume $n>k=8$. Then*

      *$cn = 31n = 30n + n > 30n+8$, so $30n+8 < cn$.*
- *Show that $n^2+1$ is $O(n^2)$.*
  - *Show $\exists c, k$: $n^2+1 \leq cn^2$, $\forall n>k$: .*
    - *Let $c=2$, $k=1$. Assume $n>1$. Then*

      *$cn^2 = 2n^2 = n^2+n^2 > n^2+1$, or $n^2+1< cn^2$.*

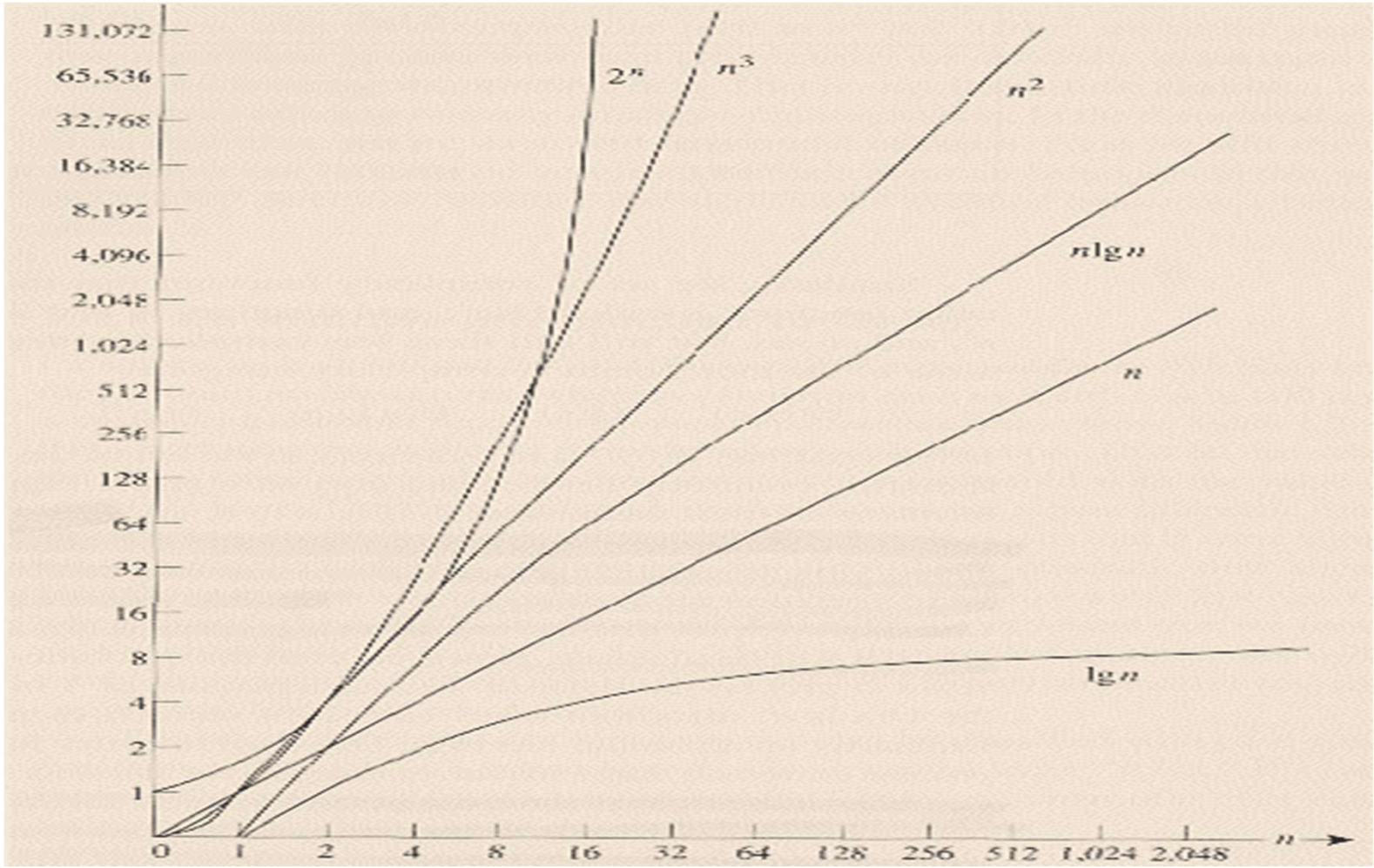
# *Big-O example, graphically*

- *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 everywhere to the right of n=8.*




$30n+8 \in O(n)$

# *Common orders of magnitude*



# Common orders of magnitude

**Table 1.4** Execution times for algorithms with the given time complexities

| $n$ | $f(n) = \lg n$ | $f(n) = n$ | $f(n) = n \lg n$ | $f(n) = n^2$ | $f(n) = n^3$ | $f(n) = 2^n$ |
|---|---|---|---|---|---|---|
| 10 | 0.003 $\mu$s* | 0.01 $\mu$s | 0.033 $\mu$s | 0.1 $\mu$s | 1 $\mu$s | $\mu$s |
| 20 | 0.004 $\mu$s | 0.02 $\mu$s | 0.086 $\mu$s | 0.4 $\mu$s | 8 $\mu$s | ms† |
| 30 | 0.005 $\mu$s | 0.03 $\mu$s | 0.147 $\mu$s | 0.9 $\mu$s | 27 $\mu$s | s |
| 40 | 0.005 $\mu$s | 0.04 $\mu$s | 0.213 $\mu$s | 1.6 $\mu$s | 64 $\mu$s | 18.3 mir |
| 50 | 0.005 $\mu$s | 0.05 $\mu$s | 0.282 $\mu$s | 2.5 $\mu$s | 25 $\mu$s | 13 days |
| $10^2$ | 0.007 $\mu$s | 0.10 $\mu$s | 0.664 $\mu$s | 10 $\mu$s | 1 ms | $4 \times 10^{15}$ years |
| $10^3$ | 0.010 $\mu$s | 1.00 $\mu$s | 9.966 $\mu$s | 1 ms | 1 s | |
| $10^4$ | 0.013 $\mu$s | 0 $\mu$s | 130 $\mu$s | 100 ms | 16.7 min | |
| $10^5$ | 0.017 $\mu$s | 0.10 ms | 1.67 ms | 10 s | 11.6 days | |
| $10^6$ | 0.020 $\mu$s | 1 ms | 19.93 ms | 16.7 min | 31.7 years | |
| $10^7$ | 0.023 $\mu$s | 0.01 s | 0.23 s | 1.16 days | 31,709 years | |
| $10^8$ | 0.027 $\mu$s | 0.10 s | 2.66 s | 115.7 days | $3.17 \times 10^7$ years | |
| $10^9$ | 0.030 $\mu$s | 1 s | 29.90 s | 31.7 years | | |

*1 $\mu$s = $10^{-6}$ second.
†1 ms = $10^{-3}$ second.

# *Order-of-Growth in Expressions*

- "O(f)" can be used as a term in an arithmetic expression .

*E.g.*: we can write "$x^2$+x+1" as "$x^2$+O($x$)" meaning "$x^2$ plus some function that is O($x$)".

- Formally, you can think of any such expression as denoting a set of functions:

$$\text{“}x^2+O(x)\text{”} :\equiv \{g \mid \exists f \in O(x): g(x) = x^2 + f(x)\}$$

# *Useful Facts about Big O*

› Constants …

$\forall c>0,\ O(cf)=O(f+c)=O(f-c)=O(f)$

› Sums:

- If $g\in O(f)$ and $h\in O(f)$, then $g+h\in O(f)$.
- If $g\in O(f_1)$ and $h\in O(f_2)$, then

$$g+h\in O(f_1+f_2)=O(\max(f_1,f_2))$$

(Very useful!)

# *More Big-O facts*

- *Products*:

  *If* $g \in O(f_1)$ *and* $h \in O(f_2)$, *then* $gh \in O(f_1 f_2)$

- *Big O, as a relation, is transitive*:

  $f \in O(g) \wedge g \in O(h) \rightarrow f \in O(h)$

# *More Big O facts*

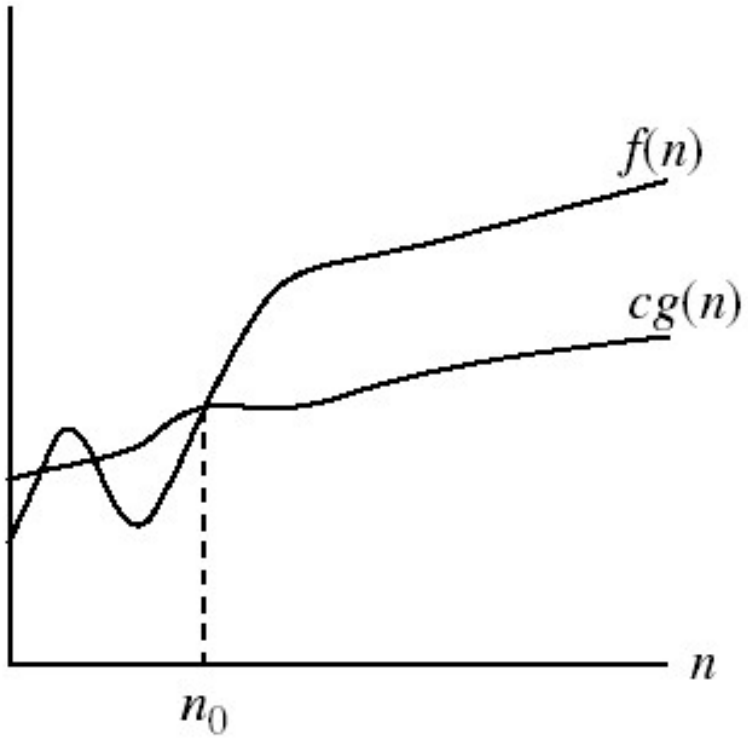
- $\forall$ $f,g$ & constants $a,b \in \mathbf{R}$, with $b \geq 0$,
  - $af = O(f)$ (*e.g.* $3x^2 = O(x^2)$)
  - $f + O(f) = O(f)$ (*e.g.* $x^2 + x = O(x^2)$)
  - $|f|^{1-b} = O(f)$ (*e.g.* $x^{-1} = O(x)$)
  - $(\log_b |f|)^a = O(f)$ (*e.g.* $\log x = O(x)$)
  - $g = O(fg)$ (*e.g.* $x = O(x \log x)$)
  - $fg \neq O(g)$ (*e.g.* $x \log x \neq O(x)$)
  - $a = O(f)$ (*e.g.* $3 = O(x)$)

## *Definition: $\Omega(g)$, at least order g*

*Let f, g be any function* $\mathbf{R} \rightarrow \mathbf{R}$.

- *We say that "f is at least order g", written* $\Omega(g)$, *if* $\exists\ \mathbf{c}, \mathbf{k}$: $\mathbf{f(x)} \geq \mathbf{cg(x)}, \forall\ \mathbf{x} > \mathbf{k}$
  - *"Beyond some point k, function f is at least a constant c times g (i.e., proportional to g)."*
  - *Often, one deals only with positive functions and can ignore absolute value symbols.*
- *"f is at least order g", or "f is* $\Omega(g)$*", or "f=* $\Omega(g)$*" all just mean that f*$\in \Omega(g)$.

# *Big- Ω Visualization*




$g(n)$ is an ***asymptotic lower bound*** for $f(n)$.

# *Definition: $\Theta(g)$, exactly order g*

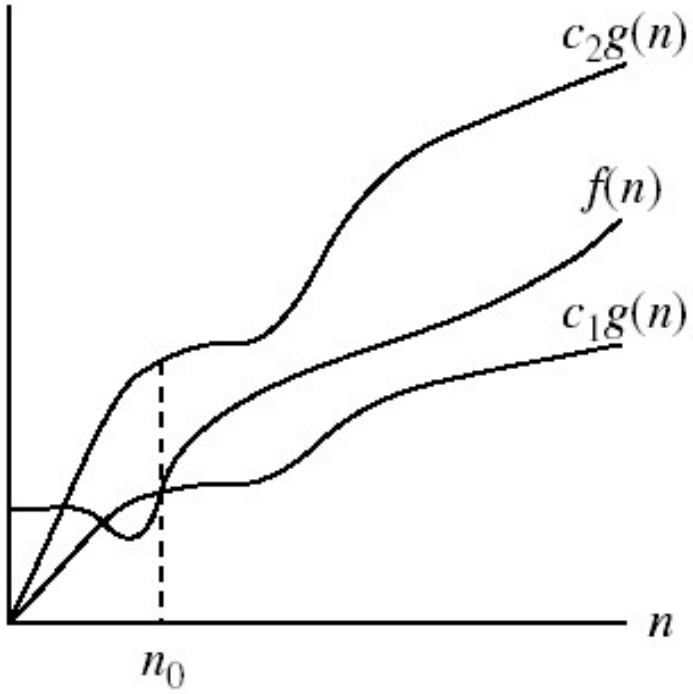
- *If $f \in O(g)$ and $g \in O(f)$ then we say "g and f are of the same order" or "f is (exactly) order g" and write $f \in \Theta(g)$.*
- *Another equivalent definition:*

$$\exists c_1 c_2, k: \; c_1 g(x) \leq f(x) \leq c_2 g(x), \; \forall x > k$$

- *"Everywhere beyond some point k, f(x) lies in between two multiples of g(x)."*
- $\Theta(g) \equiv O(g) \cap \Omega(g)$

  *(i.e., $f \in O(g)$ and $f \in \Omega(g)$ )*

# *Big- Θ Visualization*




$g(n)$ is an ***asymptotically tight bound*** for $f(n)$.

# *Rules for* $\Theta$

- *Mostly like rules for O( ), except:*
- $\forall f, g>0$ *& constants* $a,b \in \mathbf{R}$*, with* $b>0$*,*
  $af \in \Theta(f)$ ← *Same as with O.*
  $f \notin \Theta(fg)$ *unless* $g=\Theta(1)$ ← *Unlike O.*
  $|f|^{1-b} \notin \Theta(f)$*, and* ← *Unlike with O.*
  $(\log_b |f|)^c \notin \Theta(f)$*.* ← *Unlike with O.*
- *The functions in the latter two cases we say are strictly of lower order than* $\Theta(f)$*.*

# Θ *example*

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

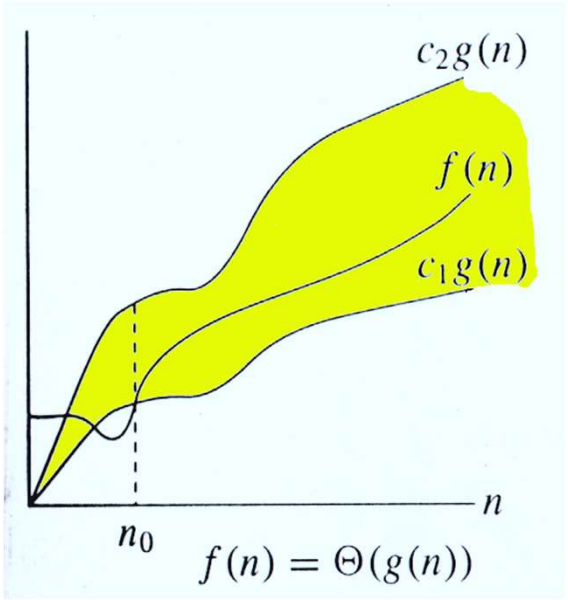
$$\left(\sum_{i=1}^{n} i\right) = n(n+1)/2$$
$$= n\,\Theta(n)/2$$
$$= n\,\Theta(n)$$
$$= \Theta(n^2)$$

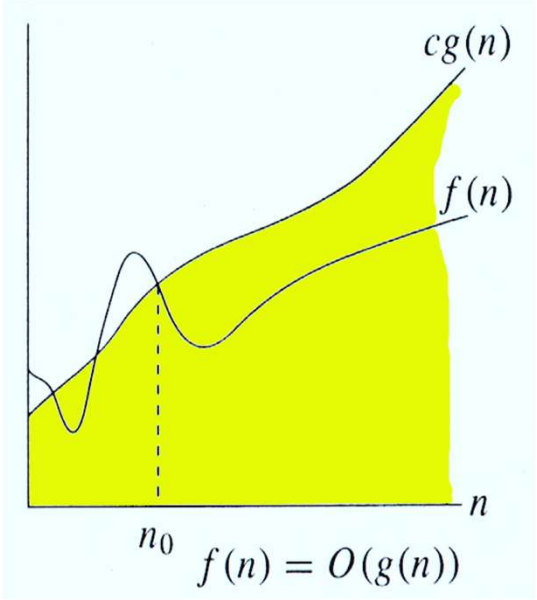
# Big O runtime of algorithm: Example

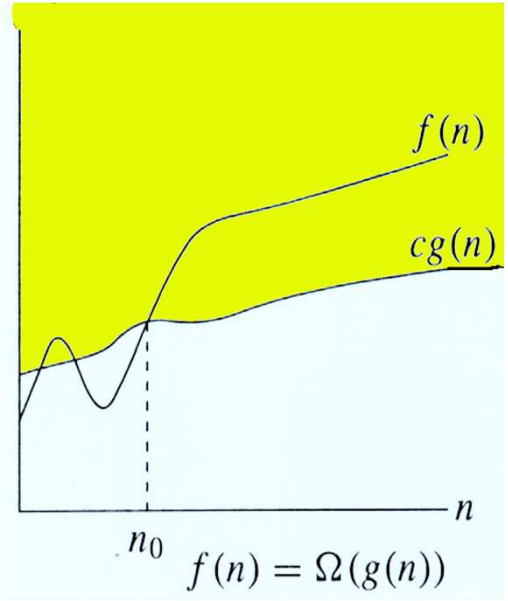
```
function findFirstIndexOfNumber(number, array) {
  for (let i = 0; i < array.length; i++) {
    if (array[i] === number) {
      return i;
    }
  }
  return -1
}
```

This algorithm needs to loop through the array until it finds the first occurrence of the number we are looking for. It doesn't matter how big the data set is, though - at most we need to loop through it once. The algorithm is O(N) linear time.

# *Relations Between* Θ, *O*, Ω



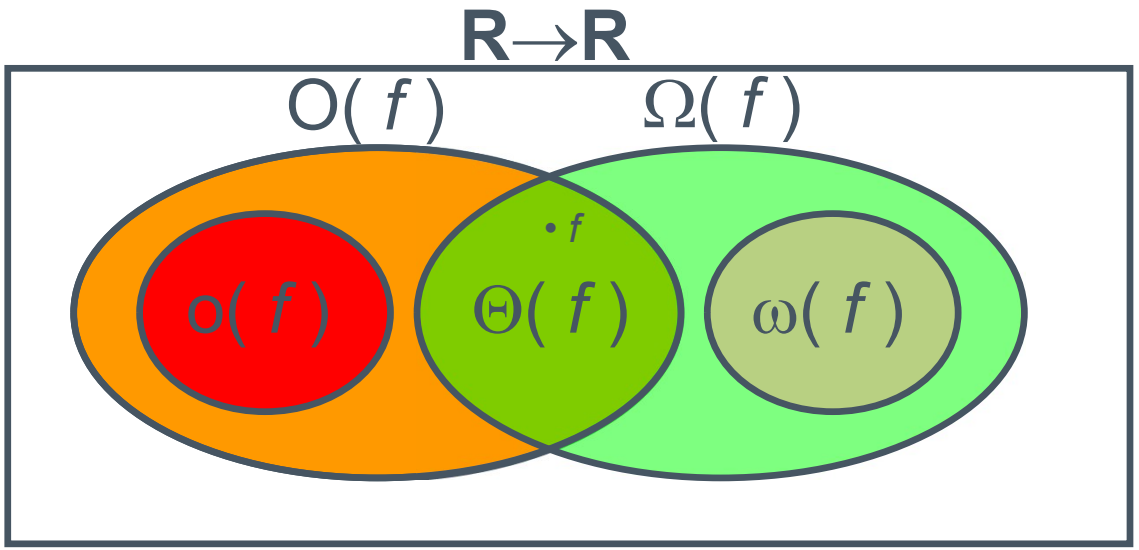
# *Other Order-of-Growth Relations*

› $o(g) = \{f \mid \forall c\, \exists k: f(x) < cg(x), \forall x > k\}$
"The functions that are *strictly lower* order than g." $o(g) \subset O(g) - \Theta(g)$.

› $\omega(g) = \{f \mid \forall c\, \exists k: cg(x) < f(x), \forall x > k\}$
"The functions that are *strictly higher* order than g." $\omega(g) \subset \Omega(g) - \Theta(g)$.

# *Relations Between the Relations*

› *Subset relations between order-of-growth sets.*



# *Review: Orders of Growth*

*Definitions of order-of-growth sets,* $\forall g : \mathbf{R} \rightarrow \mathbf{R}$

- $O(g) \equiv \{f \mid \exists\ c, k: f(x) \leq cg(x), \forall x>k\}$ *Big O*
- $o(g) \equiv \{f \mid \forall\ c \exists k: f(x) < cg(x), \forall x>k\}$ *small o*
- $\Omega(g) \equiv \{f \mid \exists\ c, k: f(x) \geq cg(x), \forall x>k\}$ *Big Omega*
- $\omega(g) \equiv \{f \mid \forall\ c \exists k: f(x) > cg(x), \forall x>k\}$ *small Omega*
- $\Theta(g) \equiv \{f \mid \exists\ c_1 c_2, k: c_1 g(x) \leq f(x) \leq c_2 g(x), \forall x>k\}$ *Theta Notation*


› Question: $T(n) = 3n^3 + 2n + 7 \in \Theta(n^3)$

- If $n \geq 1$, then $T(n) = 3n^3 + 2n + 7 \leq 3n^3 + 2n^3 + 7n^3 = 12n^3$. Hence $T(n) \in O(n^3)$.
- On the other hand, $T(n) = 3n^3 + 2n + 7 > n^3$ for all positive $n$. Therefore $T(n) \in \Omega(n^3)$.
- And consequently $T(n) \in \Theta(n^3)$.

# *Summary*

- Asymptotic analysis helps to highlight the order of growth of functions to compare algorithms
- Common asymptotic notation are of O, o, Ω, ω and Θ.
- Definitions of each notation uses the two constants k (or n0 ) and c.
- Constant k refer to the point where two function shows same value.
- While constant c refer to the value of multiple time growth of f(n) with respect to g(n).

# Thank You!!!

Have a good day