Asymptotic Notations

- Algorithms perform f(n) basic operations to accomplish task
  – Identify that function
  – Identify size of problem (n)
  – Count number of operations in terms of n

Execution time

- Time computer takes to execute $f(n)$ operations is $cf(n)$
- where c
  - depends on speed of computer and
  - varies from computer to computer

Development of Notation

- Not concerned with small values of n
- Concerned with VERY LARGE values of n
- Asymptotic – refers to study of function f as n approaches infinity

- Example: $f(n) = n^2+4n + 20$

  n^2 is the dominant term and the term $4n + 20$ becomes insignificant as n grows larger

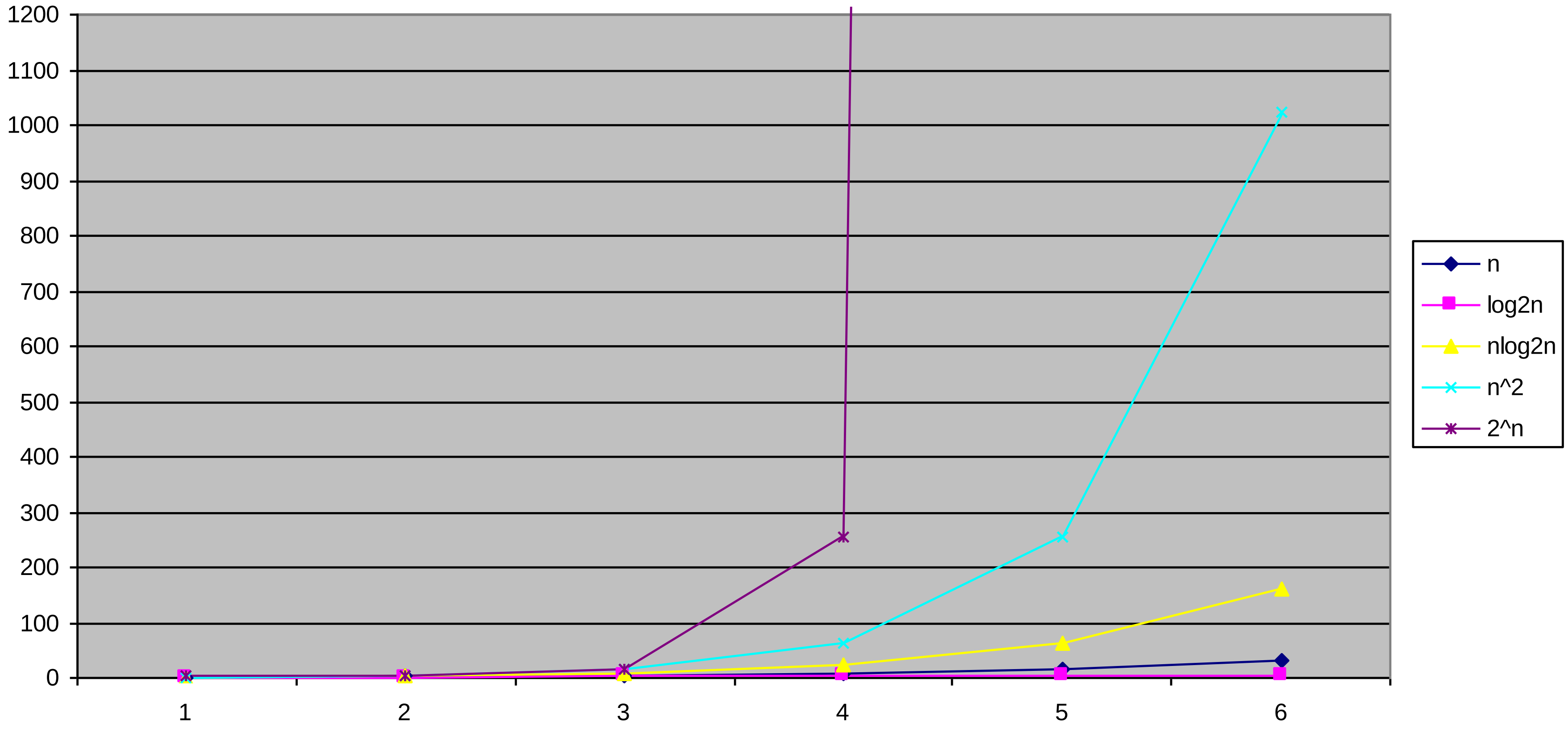
Development of Notation

- Drop insignificant terms and constants
- Say function is of $O(n^2)$ called Big-O of n^2

- Common Big-O functions in algorithm analysis
  - $g(n) = 1$ (growth is constant)
  - $g(n) = \log_2 n$ (growth is logarithmic)
  - $g(n) = n$ (growth is linear)
  - $g(n) = n \log_2 n$ (growth is faster than linear)
  - $g(n) = n^2$ (growth is quadratic)
  - $g(n) = 2^n$ (growth is exponential)

n	*log2n*	*nlog2n*	*n^2*	*2^n*
1	0	0	1	2
2	1	2	4	4
4	2	8	16	16
8	3	24	64	256
16	4	64	256	65536
32	5	160	1024	4294967296

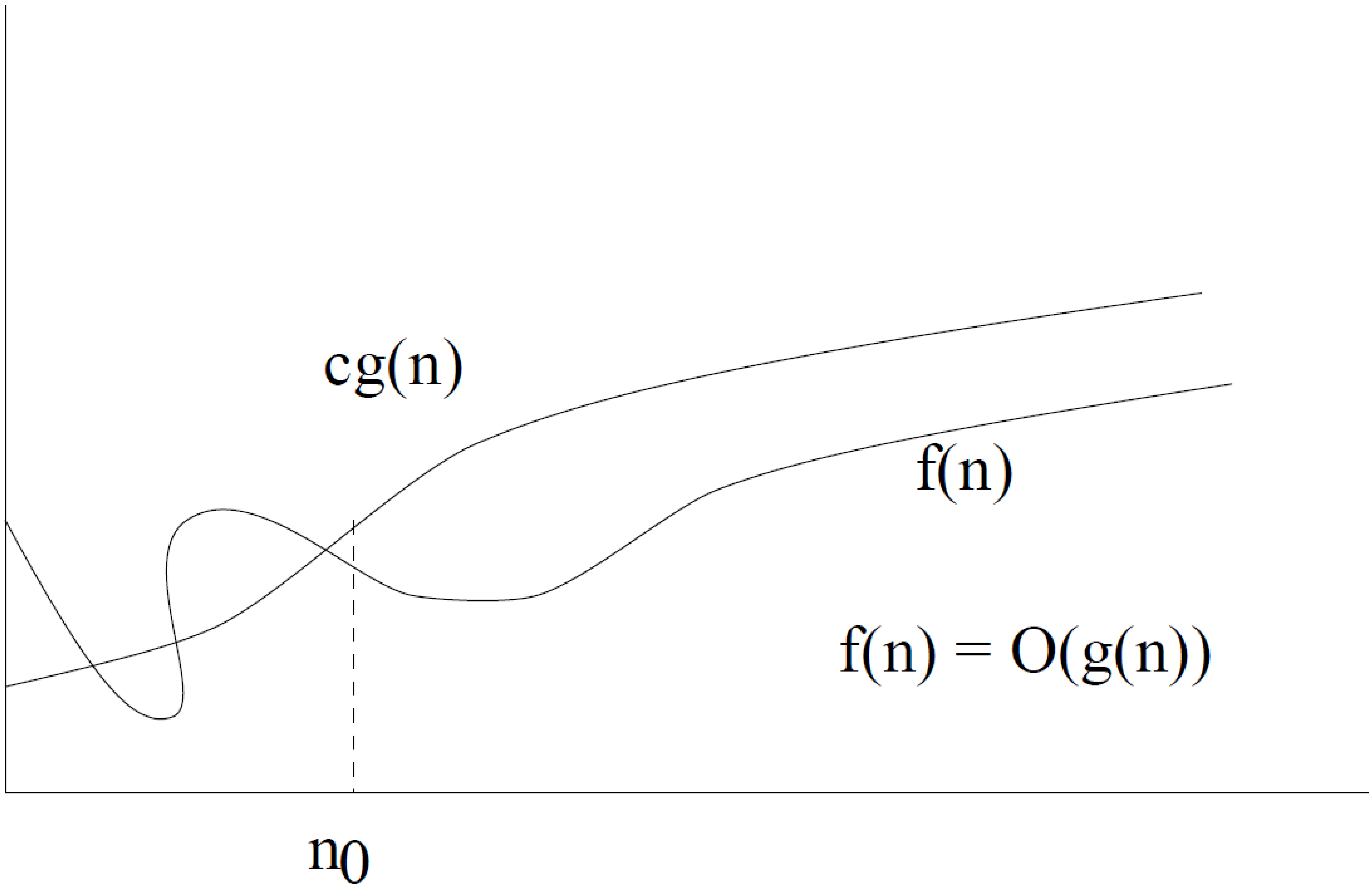
Common Growth Functions (How f(n) grows as n grows)



Big Oh

- **Definition:** $f(n) = O(g(n))$ iff there are two positive constants c and n_0 such that
$$|f(n)| \leq c\,|g(n)| \text{ for all } n \geq n_0$$
- If $f(n)$ is nonnegative, we can simplify the last condition to
$$0 \leq f(n) \leq c\,g(n) \text{ for all } n \geq n_0$$
- We say that "$f(n)$ is big-O of $g(n)$."
- As n increases, $f(n)$ grows no faster than $g(n)$. In other words, $g(n)$ is an *asymptotic upper bound* on $f(n)$.

Big Oh



Example: $n^2 + n = O(n^3)$

Proof:

- Here, we have $f(n) = n^2 + n$, and $g(n) = n^3$
- Notice that if $n \geq 1$, $n \leq n^3$ is clear.
- Also, notice that if $n \geq 1$, $n^2 \leq n^3$ is clear.
- **Side Note:** In general, if $a \leq b$, then $n^a \leq n^b$ whenever $n \geq 1$. This fact is used often in these types of proofs.

- Therefore,

$$n^2 + n \le n^3 + n^3 = 2n^3$$

- We have just shown that

$$n^2 + n \le 2n^3 \text{ for all } n \ge 1$$

- Thus, we have shown that $n^2 + n = O(n^3)$ (by definition of Big-O, with $n_0 = 1$, and $c = 2$.)

Big-Ω notation

- **Definition:** $f(n) = \Omega(g(n))$ iff there are two positive constants c and n_0 such that
$$|f(n)| \geq c\,|g(n)| \text{ for all } n \geq n_0$$
- If $f(n)$ is nonnegative, we can simplify the last condition to
$$0 \leq c\,g(n) \leq f(n) \text{ for all } n \geq n_0$$
- We say that "$f(n)$ is omega of $g(n)$."
- As n increases, $f(n)$ grows no slower than $g(n)$. In other words, $g(n)$ is an *asymptotic lower bound* on $f(n)$.

Big-Ω notation

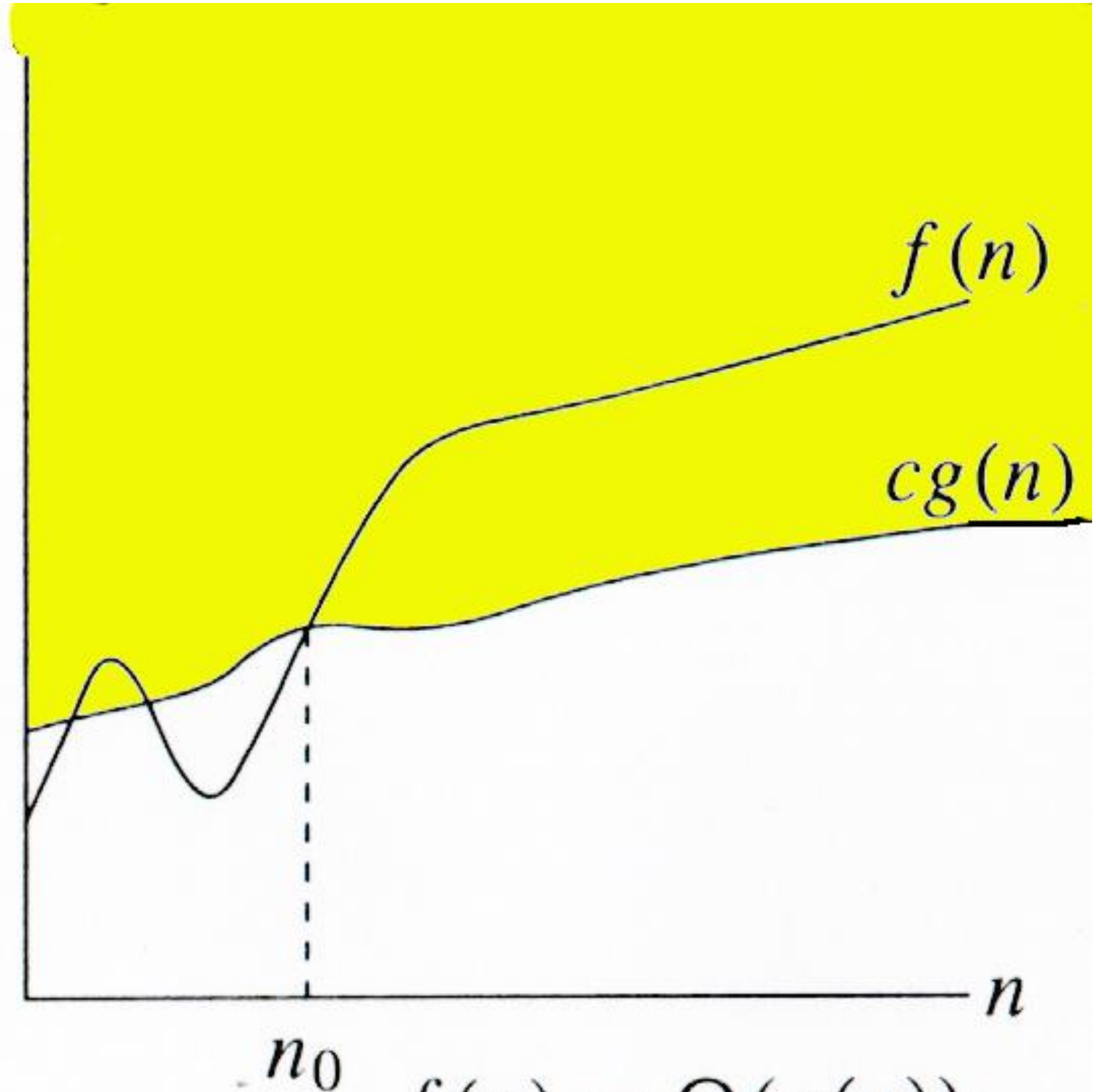
- **Definition:** $f(n) = \Omega(g(n))$ iff there are two positive constants c and n_0 such that

$$|f(n)| \geq c\,|g(n)| \text{ for all } n \geq n_0$$

- If $f(n)$ is nonnegative, we can simplify the last condition to

$$0 \leq c\,g(n) \leq f(n) \text{ for all } n \geq n_0$$

- We say that "$f(n)$ is omega of $g(n)$."
- As n increases, $f(n)$ grows no slower than $g(n)$. In other words, $g(n)$ is an *asymptotic lower bound* on $f(n)$.



$f(n) = \Omega(g(n))$

Example: $n^3 + 4n^2 = \Omega(n^2)$

Proof:

- Here, we have $f(n) = n^3 + 4n^2$, and $g(n) = n^2$
- It is not too hard to see that if $n \geq 0$,

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

- We have already seen that if $n \geq 1$,

$$n^2 \leq n^3$$

Thus when $n \geq 1$,

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

Therefore,

$$1n^2 \leq n^3 + 4n^2 \text{ for all } n \geq 1$$

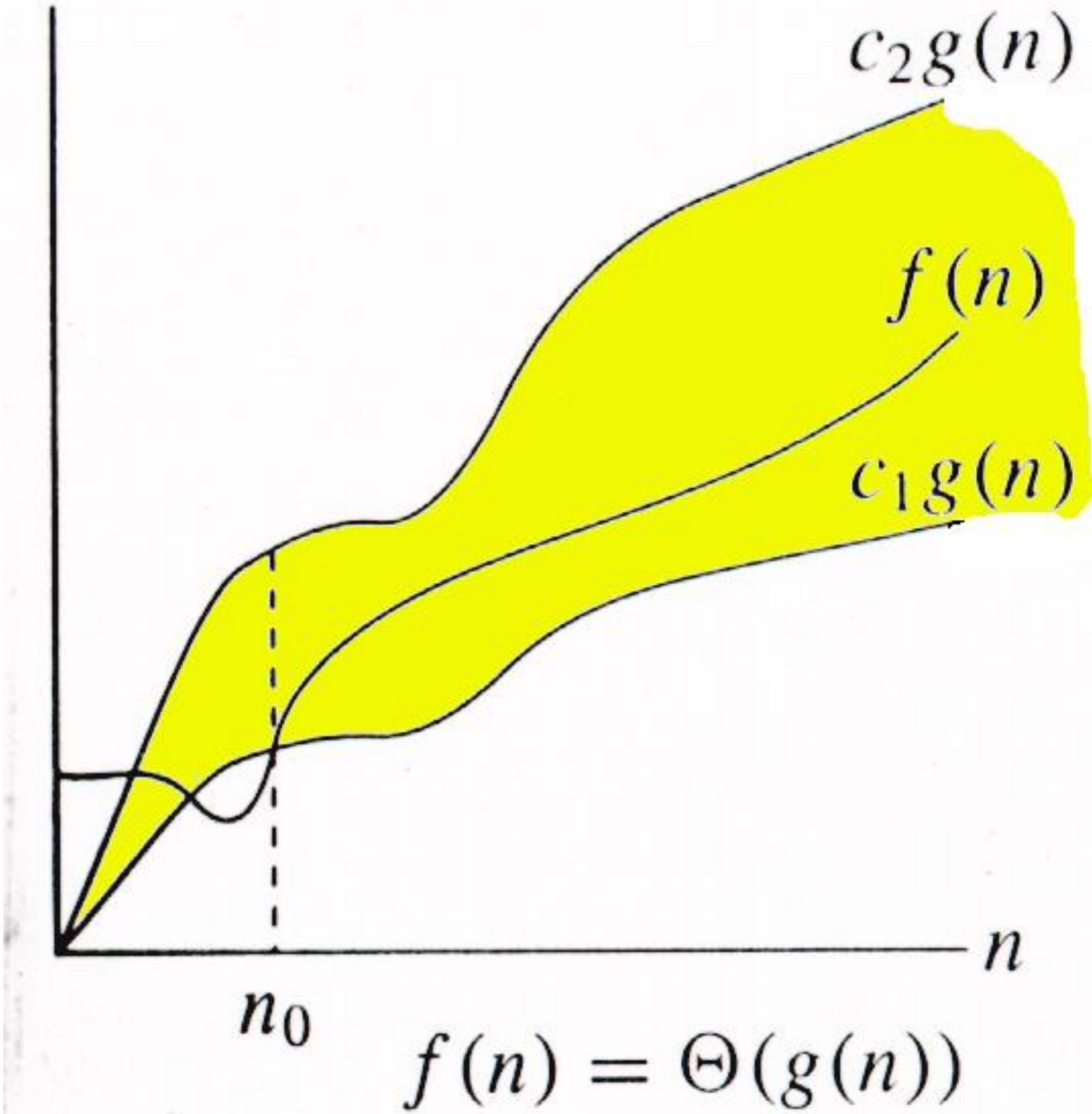
Thus, we have shown that $n^3 + 4n^2 = \Omega(n^2)$
(by definition of Big-Ω, with $n_0 = 1$, and $c = 1$.)

Big-Θ notation

- **Definition:** $f(n) = \Theta(g(n))$ iff there are three positive constants c_1, c_2 and n_0 such that
$$c_1|g(n)| \leq |f(n)| \leq c_2|g(n)| \text{ for all } n \geq n_0$$
- If $f(n)$ is nonnegative, we can simplify the last condition to
$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ for all } n \geq n_0$$
- We say that "$f(n)$ is theta of $g(n)$."
- As n increases, $f(n)$ grows at the same rate as $g(n)$. In other words, $g(n)$ is an *asymptotically tight bound* on $f(n)$.

Big-Θ notation

- **Definition:** $f(n) = \Theta(g(n))$ iff there are three positive constants c_1, c_2 and n_0 such that
$$c_1|g(n)| \leq |f(n)| \leq c_2|g(n)| \text{ for all } n \geq n_0$$
- If $f(n)$ is nonnegative, we can simplify the last condition to
$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ for all } n \geq n_0$$
- We say that "$f(n)$ is theta of $g(n)$."
- As n increases, $f(n)$ grows at the same rate as $g(n)$. In other words, $g(n)$ is an *asymptotically tight bound* on $f(n)$.



$c_2 g(n)$
$f(n)$
$c_1 g(n)$
n
n_0
$f(n) = \Theta(g(n))$

Example: $n^2 + 5n + 7 = \Theta(n^2)$

Proof:

- When $n \geq 1$,

$$n^2 + 5n + 7 \leq n^2 + 5n^2 + 7n^2 \leq 13n^2$$

- When $n \geq 0$,

$$n^2 \leq n^2 + 5n + 7$$

- Thus, when $n \geq 1$

$$1n^2 \leq n^2 + 5n + 7 \leq 13n^2$$

Thus, we have shown that $n^2 + 5n + 7 = \Theta(n^2)$ (by definition of Big-Θ, with $n_0 = 1$, $c_1 = 1$, and $c_2 = 13$.)

Arithmetic of Big-O, Ω, and Θ notations

- Transitivity:
  - $f(n) \in O(g(n))$ and $g(n) \in O(h(n)) \Rightarrow f(n) \in O(h(n))$
  - $f(n) \in \Theta(g(n))$ and $g(n) \in \Theta(h(n)) \Rightarrow f(n) \in \Theta(h(n))$
  - $f(n) \in \Omega(g(n))$ and $g(n) \in \Omega(h(n)) \Rightarrow f(n) \in \Omega(h(n))$

Arithmetic of Big-O, Ω, and Θ notations

- Scaling: if $f(n) \in O(g(n))$ then for any $k > 0$, $f(n) \in O(kg(n))$

- Sums: if $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$ then $(f_1 + f_2)(n) \in O(max(g_1(n), g_2(n)))$

Prove that

$$5n^2 + 3n + 20 = O(n^2)$$

$$\frac{1}{2}n^2 + 3n = \Theta(n^2)$$

$$(n \log n - 2\,n + 13) = \Omega(n \log n)$$

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

Show that $\frac{1}{2}n^2 + 3n = \Theta(n^2)$

Proof:

- Notice that if $n \geq 1$,

$$\frac{1}{2}n^2 + 3n \leq \frac{1}{2}n^2 + 3n^2 = \frac{7}{2}n^2$$

- Thus,

$$\frac{1}{2}n^2 + 3n = O(n^2)$$

- Also, when $n \geq 0$,

Also, when $n \geq 0$,

$$\frac{1}{2}n^2 \leq \frac{1}{2}n^2 + 3n$$

So

$$\frac{1}{2}n^2 + 3n = \Omega(n^2)$$

Since $\frac{1}{2}n^2 + 3n = O(n^2)$ and $\frac{1}{2}n^2 + 3n = \Omega(n^2)$,

$$\frac{1}{2}n^2 + 3n = \Theta(n^2)$$

Show that $(n \log n - 2n + 13) = \Omega(n \log n)$

Proof: We need to show that there exist positive constants c and n_0 such that

$$0 \le c\,n \log n \le n \log n - 2n + 13 \text{ for all } n \ge n_0.$$

Since $\quad n \log n - 2n \le n \log n - 2n + 13,$

we will instead show that

$$c\,n \log n \le n \log n - 2n,$$

which is equivalent to

$$c \le 1 - \frac{2}{\log n}, \text{ when } n > 1.$$

If $n \ge 8$, then $2/(\log n) \le 2/3$, and picking $c = 1/3$ suffices. Thus if $c = 1/3$ and $n_0 = 8$, then for all $n \ge n_0$, we have

$$0 \le c\,n \log n \le n \log n - 2n \le n \log n - 2n + 13.$$

Thus $(n \log n - 2n + 13) = \Omega(n \log n)$.

Show that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$

Proof:

- We need to find positive constants c_1, c_2, and n_0 such that

$$0 \leq c_1 n^2 \leq \frac{1}{2}n^2 - 3\,n \leq c_2 n^2 \text{ for all } n \geq n_0$$

- Dividing by n^2, we get

$$0 \leq c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

- $c_1 \leq \frac{1}{2} - \frac{3}{n}$ holds for $n \geq 10$ and $c_1 = 1/5$
- $\frac{1}{2} - \frac{3}{n} \leq c_2$ holds for $n \geq 10$ and $c_2 = 1$.
- Thus, if $c_1 = 1/5$, $c_2 = 1$, and $n_0 = 10$, then for all $n \geq n_0$,

$$0 \leq c_1 n^2 \leq \frac{1}{2}n^2 - 3\,n \leq c_2 n^2 \text{ for all } n \geq n_0.$$

Thus we have shown that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$.

o-notation

For a given function $g(n)$, the set little-o:

$o(g(n)) = \{f(n): \forall\, c > 0, \exists\, n0 > 0$ such that $\forall\, n \geq n0$, we have $0 \leq f(n) < cg(n)\}$.

$f(n)$ becomes insignificant relative to $g(n)$ as n approaches infinity:

$$\lim_{n \to \infty} [f(n) / g(n)] = 0$$

$g(n)$ is an ***upper bound*** for $f(n)$ that is not asymptotically tight.

Observe the difference in this definition from previous ones. **Why?**

ω -notation

For a given function $g(n)$, the set little-omega:

w(g(n)) = {f(n): $\forall$ c > 0, $\exists$ n0 > 0 such that
$\forall$ n $\geq$ n0, we have 0 $\leq$ cg(n) < f(n)}.

$f(n)$ becomes arbitrarily large relative to $g(n)$ as n approaches infinity:

$$\lim_{n \to \infty} [f(n) / g(n)] = \infty.$$

$g(n)$ is a ***lower bound*** for $f(n)$ that is not asymptotically tight.

Comparison of Functions

$$f \leftrightarrow g \ \approx \ a \leftrightarrow b$$

$$f(n) = O(g(n)) \ \approx \ a \le b$$

$$f(n) = \Omega(g(n)) \ \approx \ a \ge b$$

$$f(n) = \Theta(g(n)) \ \approx \ a = b$$

$$f(n) = o(g(n)) \ \approx \ a < b$$

$$f(n) = \omega(g(n)) \ \approx \ a > b$$

True False $3n + 10\, n \log n = O(n \log n)$

True False $3\, n + 10\, n \log n = \Omega\, (n \log n)$

True False $3\, n + 10\, n \log n = \Theta(n \log n)$

True False $n \log n + \frac{n}{2} = O(n^{2} \log n)$

True False $5n^2 + n = \Theta\, (n^{3})$

True False $2^n + n^2 = \Omega\, (1)$