CSE 250: Asymptotic Analysis

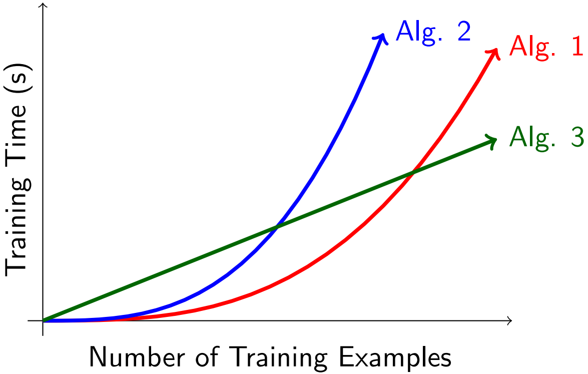
Lecture 5

Sept 6, 2024

Reminders

- AI Quiz due Sun, Sept 8 at 11:59 PM.
  - Your final submission must have a score of 1.0 to pass the class.
  - If you can't submit in autolab, let course staff know ASAP.
- PA0 due Sun, Sept 8 at 11:59 PM.
  - All you need to do is make sure you have a working environment.
  - If you can't submit in autolab, let course staff know ASAP.
- WA1 due Sun, Sept 8 at 11:59 PM.
  - Summations, Limits, Exponentials; Friday's Lecture

Runtime



How many Steps?

```
x = 10;
```

vs

```
x = 10;
y = x + 1;
```

1 java instruction vs 2 java instructions

How many Steps?

```
0: bipush        10
2: istore_1
3: return
```

vs

```
0: bipush        10
2: istore_1
3: iload_1
4: iconst_1
5: iadd
6: istore_2
7: return
```

3 java bytecode instructions vs 7 java bytecode instructions

title

```
x = 10;
```

vs

```
x = 10;
y = x + 1;
```

$\theta(1)$ vs $\theta(1)$ (Both code snippets take 'constant' time).

Steps

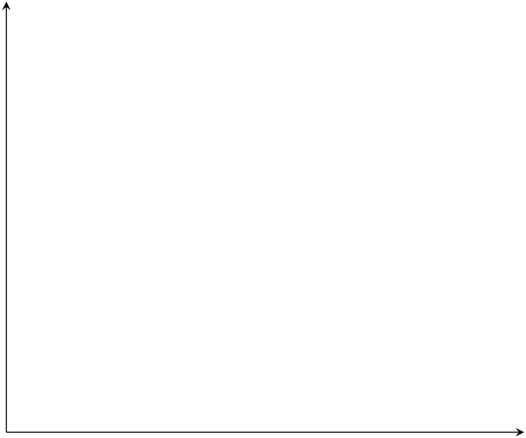
$\theta(1)$ is any computation that always[1] has the same runtime.

[1] Offer void where prohibited, some approximations may apply.

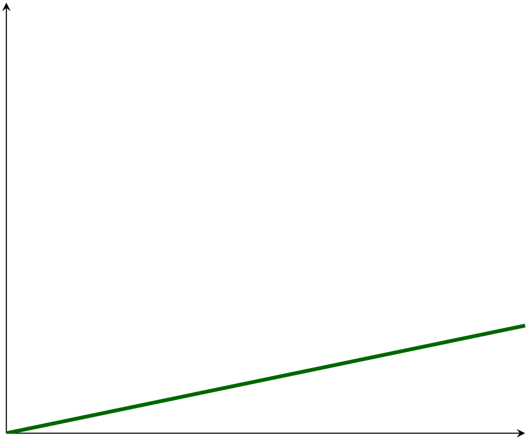
Class Names

- $\theta(1)$: Constant
- $\theta(\log(N))$: Logarithmic
- $\theta(N)$: Linear
- $\theta(N \log(N))$: Log-Linear
- $\theta(N^2)$: Quadratic
- $\theta(N^k)$ (for any $k \geq 1$): Polynomial
- $\theta(2^N)$: Exponential

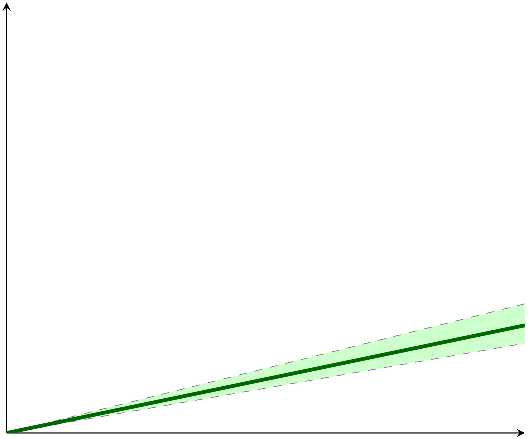
Complexity Classes



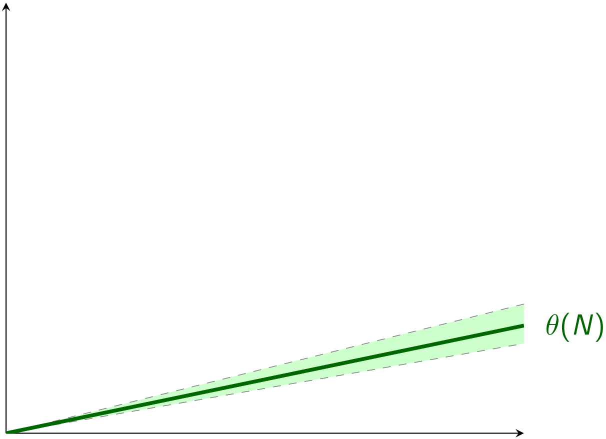
Complexity Classes



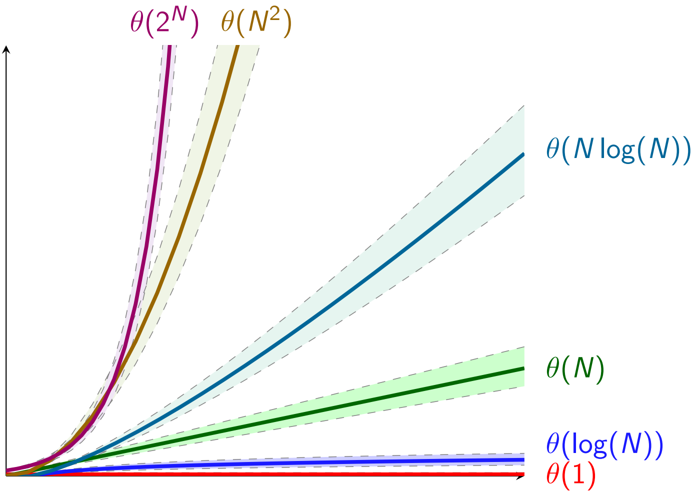
Complexity Classes



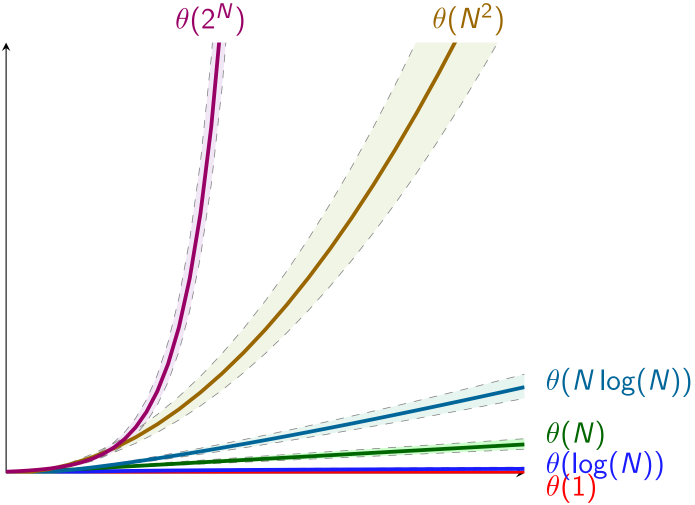
Complexity Classes



Complexity Classes



Zoom Out



Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Pick Algorithm 2

Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Pick Algorithm 2 . . . usually.

Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Pick Algorithm 2 ... usually.

1. Algorithm 1 is $\theta(N)$
2. Algorithm 2 is $\theta(N)$

Comparing Algorithms

1. Algorithm 1 is $\theta(N^2)$
2. Algorithm 2 is $\theta(N)$

Pick Algorithm 2 . . . usually.

1. Algorithm 1 is $\theta(N)$
2. Algorithm 2 is $\theta(N)$

Measure actual runtimes

Baseline

If $g(N) = c_1 + c_2 f(N)$, then g, f are in the same complexity class.

Complexity Bounds

For $N > 1$:

$$N^2 \leq \quad N^2 + N \quad \leq 2N^2$$

Complexity Bounds

For $N > 1$:

$$N^2 \leq \quad N^2 + N \quad \leq 2N^2$$

$N^2 + N$ should probably be in $\theta(N^2)$ too.

Complexity Bounds

if:

- $f_{low}(N), f_{high}(N) \in \theta(g(N))$
- $f_{low}(N) \leq T(N) \leq f_{high}(N)$ (for all big enough N)

...then $T(N) \in \theta(g(N))$ too!

Complexity Bounds

f and g are in the same complexity class if:

Complexity Bounds

f and g are in the same complexity class if:

- g is bounded from above by something f-shaped

Complexity Bounds

f and g are in the same complexity class if:

- g is bounded from above by something f-shaped
- g is bounded from below by something f-shaped

Complexity Bounds

f and g are in the same complexity class if:

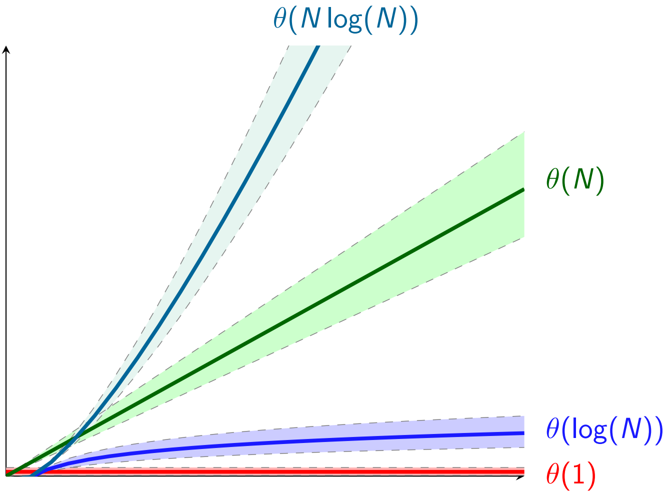
- g is bounded from above by something f-shaped
  $g(N) \in O(f(N))$
- g is bounded from below by something f-shaped

Complexity Bounds

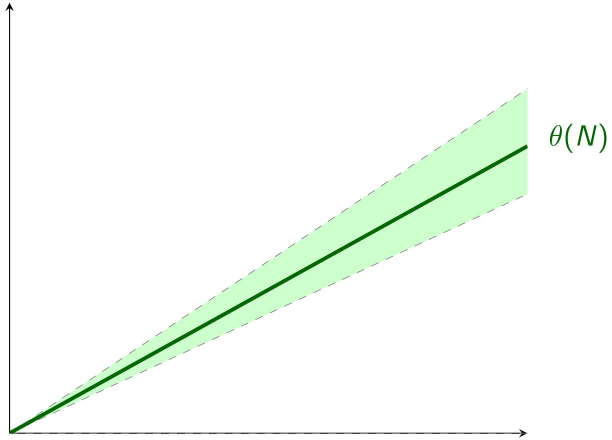
f and g are in the same complexity class if:

- g is bounded from above by something f-shaped
  $g(N) \in O(f(N))$
- g is bounded from below by something f-shaped
  $g(N) \in \Omega(f(N))$

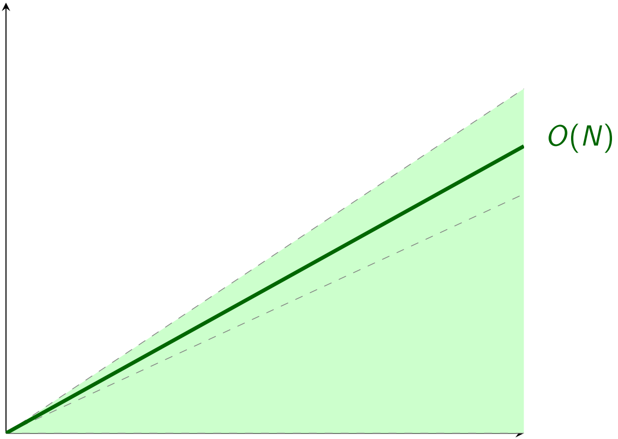
Complexity Bounds



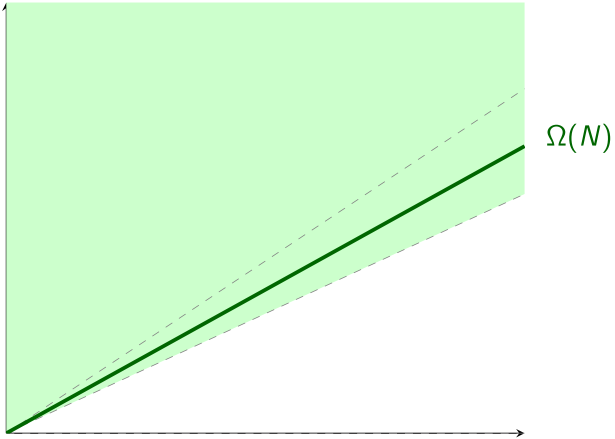
Complexity Bounds



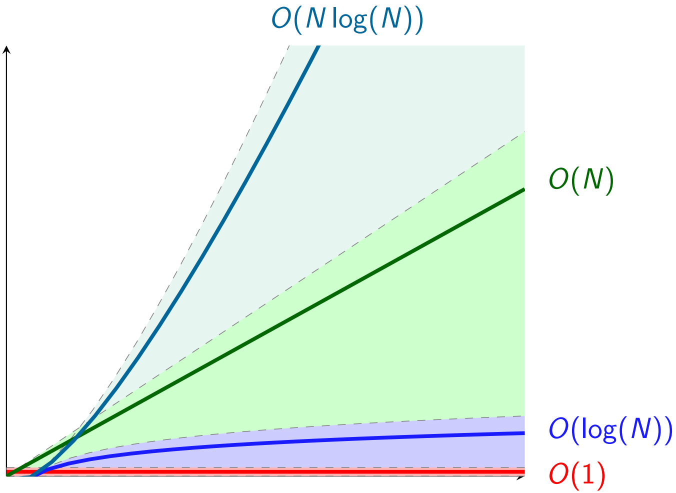
Complexity Bounds



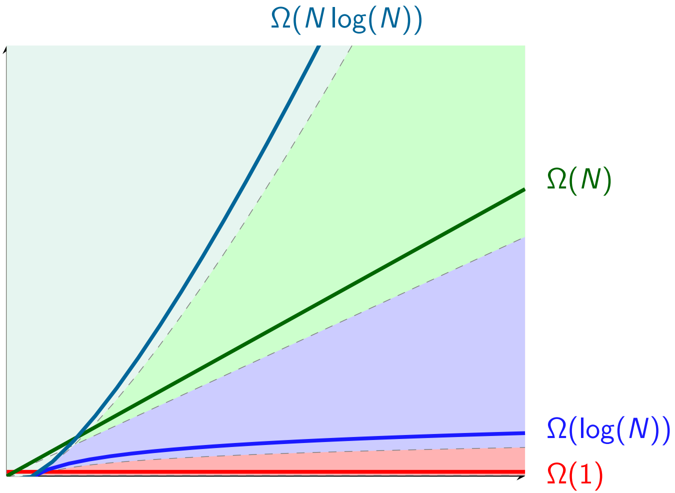
Complexity Bounds



Complexity Bounds



Complexity Bounds



Complexity Bounds

- $O(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'slower-growing' complexity classes
- $\Omega(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'faster-growing' complexity classes

Complexity Bounds

- $O(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'smaller' complexity classes
- $\Omega(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'bigger' complexity classes

Complexity Bounds

- $O(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'smaller' complexity classes
- $\Omega(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'bigger' complexity classes

$$O(f(N)) \cap \Omega(f(N)) = ???$$

Complexity Bounds

- $O(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'smaller' complexity classes
- $\Omega(f(N))$ includes:
  - All functions in $\theta(f(N))$
  - All functions in 'bigger' complexity classes

$$O(f(N)) \cap \Omega(f(N)) = \theta(f(N))$$

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if...

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > 0$: $g(N) \leq f(N)$

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > 0$: $g(N) \leq f(N)$

But what about...

- $g(N) = 2N$ **vs** $f(N) = N$ **on** any N

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > 0$: $g(N) \leq c \cdot f(N)$

But what about...

- $g(N) = 2N$ **vs** $f(N) = N$ **on** any N

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > 0$: $g(N) \leq c \cdot f(N)$

Where...

- ... there is some $c > 0$

But what about...

- $g(N) = 2N$ **vs** $f(N) = N$ **on** any N

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > 0$: $g(N) \leq c \cdot f(N)$

Where...

- ... there is some $c > 0$

But what about...

- $g(N) = 2N$ **vs** $f(N) = N$ **on** any N
- $g(N) = \log\left(\frac{N}{2}\right)$ **vs** $f(N) = N$ **on** $N = 1$

Bounding From Above

$g(N)$ is smaller than or equal to $f(N)$ if... $(g(N) \in O(f(N)))$

For all $N > N_0$: $g(N) \leq c \cdot f(N)$

Where...

- ... there is some $c > 0$
- ... there is some $N_0 > 0$

But what about...

- $g(N) = 2N$ **vs** $f(N) = N$ **on** any N
- $g(N) = \log\left(\frac{N}{2}\right)$ **vs** $f(N) = N$ **on** $N = 1$

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$
- Simplify it and show me that you get a trivially true inequality.

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$
- Simplify it and show me that you get a trivially true inequality.

To prove that $g(N)$ is **not** smaller than or equal to $f(N)$...

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$
- Simplify it and show me that you get a trivially true inequality.

To prove that $g(N)$ is **not** smaller than or equal to $f(N)$...

- $\forall N > N_0 : g(N) \leq c \cdot f(N)$

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$
- Simplify it and show me that you get a trivially true inequality.

To prove that $g(N)$ is **not** smaller than or equal to $f(N)$...

- $\forall N > N_0 : g(N) \leq c \cdot f(N)$
  $\forall N > N_0 : \frac{g(N)}{f(N)} \leq c$

Bounding from Above

To prove that $g(N)$ is smaller than or equal to $f(N)$...

- Give me a $c > 0$.
- Give me a $N_0 \geq 0$.
- Plug them into $\forall N > N_0 : g(N) \leq c \cdot f(N)$
- Simplify it and show me that you get a trivially true inequality.

To prove that $g(N)$ is **not** smaller than or equal to $f(N)$...

- $\forall N > N_0 : g(N) \leq c \cdot f(N)$
  $\forall N > N_0 : \frac{g(N)}{f(N)} \leq c$
- For any c that I get to pick...
  show me how to find an N where $\frac{g(N)}{f(N)} > c$

Chain Rule

If $X \geq Y$, $Y \geq Z$, then $X \geq Z$

To show: $X \geq Z$, find a Y and show:

- $X \geq Y$
- $Y \geq Z$

Decomposition

If $A \geq C$ and $B \geq D$ then $A + B \geq C + D$
To show $A + B \geq C + D$, show that:

- $A \geq C$
- $B \geq D$

Examples

$$g(N) = 1 \qquad f(N) = N$$

Examples

$$g(N) = 1 \qquad f(N) = N$$

$$1 \overset{?}{\leq} c \cdot N$$

Examples

$$g(N) = 1 \qquad f(N) = N$$

$$1 \overset{?}{\leq} c \cdot N$$

Is there a $c > 0$ and $N_0 > 0$ you can plug in to make this equation true for all $N \geq N_0$?

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

$$N + 2N^2 \overset{?}{\leq} c \cdot N^2$$

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

$$N + 2N^2 \overset{?}{\leq} c \cdot N^2$$
$$1 + 2N \overset{?}{\leq} c \cdot N$$

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

$$N + 2N^2 \overset{?}{\leq} c \cdot N^2$$
$$1 + 2N \overset{?}{\leq} c \cdot N$$
$$1 + 2N \overset{?}{\leq} (a + b) \cdot N$$

Define $c = a + b$

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

$$N + 2N^2 \stackrel{?}{\leq} c \cdot N^2$$

$$1 + 2N \stackrel{?}{\leq} c \cdot N$$

$$1 + 2N \stackrel{?}{\leq} (a + b) \cdot N$$

$$1 \stackrel{?}{\leq} a \cdot N$$

$$2N \stackrel{?}{\leq} b \cdot N$$

Define $c = a + b$

Examples

$$g(N) = N + 2N^2 \qquad f(N) = N^2$$

$$N + 2N^2 \overset{?}{\leq} c \cdot N^2$$

$$1 + 2N \overset{?}{\leq} c \cdot N$$

$$1 + 2N \overset{?}{\leq} (a + b) \cdot N$$

$$1 \overset{?}{\leq} a \cdot N$$

$$2N \overset{?}{\leq} b \cdot N$$

$$2 \overset{?}{\leq} b$$

Define $c = a + b$

Examples

$$1 \overset{?}{\leq} a \cdot N \tag{1}$$

$$2 \overset{?}{\leq} b \tag{2}$$

Is there an $a + b = c > 0$ and $N_0 > 0$ you can plug in to make this equation true for all $N \geq N_0$?

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

$$3N + 1 \overset{?}{\leq} c \cdot N^2$$

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

$$3N + 1 \overset{?}{\leq} c \cdot N^2$$

$$3 + \frac{1}{N} \overset{?}{\leq} c \cdot N$$

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

$$3N + 1 \overset{?}{\leq} c \cdot N^2$$

$$3 + \frac{1}{N} \overset{?}{\leq} c \cdot N$$

If $X < Y$ and $Y < Z$, then $X < Z$:

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

$$3N + 1 \stackrel{?}{\leq} c \cdot N^2$$

$$3 + \frac{1}{N} \stackrel{?}{\leq} c \cdot N$$

If $X < Y$ and $Y < Z$, then $X < Z$:

$$3 + \frac{1}{N} \leq Y \stackrel{?}{\leq} c \cdot N$$

Examples

$$g(N) = 3N + 1 \qquad f(N) = N^2$$

$$3N + 1 \overset{?}{\leq} c \cdot N^2$$

$$3 + \frac{1}{N} \overset{?}{\leq} c \cdot N$$

If $X < Y$ and $Y < Z$, then $X < Z$:

$$3 + \frac{1}{N} \leq Y \overset{?}{\leq} c \cdot N$$

$$3 + \frac{1}{N} \leq 3 + 1 \overset{?}{\leq} c \cdot N$$

Examples

$$3 + \frac{1}{N} \leq 4 \stackrel{?}{\leq} c \cdot N$$

Examples

$$3 + \frac{1}{N} \leq 4 \stackrel{?}{\leq} c \cdot N$$

Is there a $c > 0$ and $N_0 \geq 1$ you can plug in to make this equation true for all $N \geq N_0$?

Examples

$$g(N) = 1 \qquad f(N) = N^2$$

Examples

$$g(N) = 1 \qquad f(N) = N^2$$

$$1 \overset{?}{\leq} \ c \cdot N^2$$

Examples

$$g(N) = 1 \qquad f(N) = N^2$$

$$1 \stackrel{?}{\leq} c \cdot N^2$$

Is there a $c > 0$ and $N_0 > 0$ you can plug in to make this equation true for all $N \geq N_0$?

Examples

$$g(N) = 1 \qquad f(N) = N^2$$

$$1 \overset{?}{\leq} c \cdot N^2$$

Is there a $c > 0$ and $N_0 > 0$ you can plug in to make this equation true for all $N \geq N_0$?

$$1 \in O(N^2)$$

Examples

$$g(N) = 1 \qquad f(N) = N^2$$

$$1 \overset{?}{\leq} c \cdot N^2$$

Is there a $c > 0$ and $N_0 > 0$ you can plug in to make this equation true for all $N \geq N_0$?

$$1 \in O(N^2)$$

$O(f(N))$ is every mathematical function in the complexity class of $f(N)$ or a lesser class.

Tight Bounds

So... along those lines: $N \in O(N^2)$

Tight Bounds

So... along those lines: $N \in O(N^2)$
We call this a **loose** bound.

Tight Bounds

So... along those lines: $N \in O(N^2)$
We call this a **loose** bound.
$g(N) \in O(f(N))$ is a **tight** bound if there is no $f'(N)$ in a **smaller** complexity class where $g(N) \in O(f'(N))$.

Bounding From Below

$g(N) \in \Omega(f(N))$ if:

- There is some $N_0 > 0$
- There is some $c > 0$
- For all $N > N_0$: $g(N) \geq c \cdot f(N)$

Bounding From Below

$g(N) \in \Omega(f(N))$ if:

- There is some $N_0 > 0$
- There is some $c > 0$
- For all $N > N_0$: $g(N) \geq c \cdot f(N)$

$\Omega(f(N))$ is every mathematical function in the complexity class of $f(N)$ or a greater class.

Rules of Thumb

$\theta(1)$: Constant
$< \theta(\log(N))$: Logarithmic
$< \theta(N)$: Linear
$< \theta(N \log(N))$: Log-Linear
$< \theta(N^2)$: Quadratic
$< \theta(2^N)$: Exponential

Rules of Thumb

$$O(1) \subset O(\log(N))$$

Rules of Thumb

$$O(1) \subset O(\log(N))$$

$$O(\log(N)) \subset O(N)$$

Rules of Thumb

$$O(1) \subset O(\log(N))$$

$$O(\log(N)) \subset O(N)$$

$$O(N) \subset O(N \log(N))$$

Rules of Thumb

$$O(1) \subset O(\log(N))$$

$$O(\log(N)) \subset O(N)$$

$$O(N) \subset O(N \log(N))$$

$$O(N) \subset O(N^2)$$

Rules of Thumb

$$O(1) \subset O(\log(N))$$

$$O(\log(N)) \subset O(N)$$

$$O(N) \subset O(N \log(N))$$

$$O(N) \subset O(N^2)$$

...

Rules of Thumb

- $O(f(N))$ (Big-O): The complexity class of $f(N)$ and every lesser class.
- $\theta(f(N))$ (Big-θ): The complexity class of $f(N)$.
- $\Omega(f(N))$ (Big-Ω): The complexity class of $f(N)$ and every greater class.

Rules of Thumb

$$\theta$$

Rules of Thumb

$$O$$

$$\theta$$

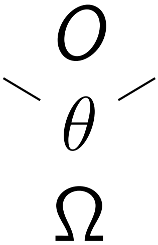
Rules of Thumb

$$O$$

$$\theta$$

$$\Omega$$

Rules of Thumb



© Aleksandra Patrzalek, 2012

Rules of Thumb

$$F(N) = f_1(N) + f_2(N) + \ldots + f_k(N)$$

What complexity class is $F(N)$ in?

Rules of Thumb

$$F(N) = f_1(N) + f_2(N) + \ldots + f_k(N)$$

What complexity class is $F(N)$ in?

$f_1(N) + f_2(N)$ is in the greater of $\theta(f_1(N))$ and $\theta(f_2(N))$.

Rules of Thumb

$$F(N) = f_1(N) + f_2(N) + \ldots + f_k(N)$$

What complexity class is $F(N)$ in?

$f_1(N) + f_2(N)$ is in the greater of $\theta(f_1(N))$ and $\theta(f_2(N))$.

$F(N)$ is in the greatest of any $\theta(f_i(N))$

Rules of Thumb

$$F(N) = f_1(N) + f_2(N) + \ldots + f_k(N)$$

What complexity class is $F(N)$ in?

$f_1(N) + f_2(N)$ is in the greater of $\theta(f_1(N))$ and $\theta(f_2(N))$.

$F(N)$ is in the greatest of any $\theta(f_i(N))$

We say the biggest f_i is the dominant term.

Algorithms at 50k-ft

- Algorithm 1 is $\theta(N^2)$
- Algorithm 2 is $\theta(N \log(N))$

Which do you pick?

Scaling Up

At $\frac{1}{4}$ ns per 'step' (4 GHz):

$f(n)$	10	20	50	100	1000
$\log(\log(n))$	0.43 ns	0.52 ns	0.62 ns	0.68 ns	0.82 ns
$\log(n)$	0.83 ns	1.01 ns	1.41 ns	1.66 ns	2.49 ns
n	2.5 ns	5 ns	12.5 ns	25 ns	0.25 μs
$n\log(n)$	8.3 ns	22 ns	71 ns	0.17 μs	2.49 μs
n^2	25 ns	0.1 μs	0.63 μs	2.5 μs	0.25 ms
n^5	25 μs	0.8 ms	78 ms	2.5 s	2.9 days
2^n	0.25 μs	0.26 ms	3.26 days	1013 years	10284 years
$n!$	0.91 ms	19 years	1047 years	10141 years	[yeah, no]

Asymptotic Notation

Big-θ (and Big-O, Big-Ω) gives us an easy shorthand for how "good" an algorithm is.