CS3500: Object-Oriented Design Spring 2014

Class 10

2.11.2014

Northeastern University
College *of* Computer and Information Science

Today...

- Assignments
- Binary Search Tree Review
- Asymptotic notation
- Efficiency
- Optimization

Readings

Please read the following paper for class on Friday, February 21, 2014:

> Chris Okasaki, "Red-black trees in a functional setting," Journal of Functional Programming, 9(4), pages 474-477, July 1999.

Link to paper on course website.

Assignments 4

- Grades posted
- Use comments to improve Assignment 5

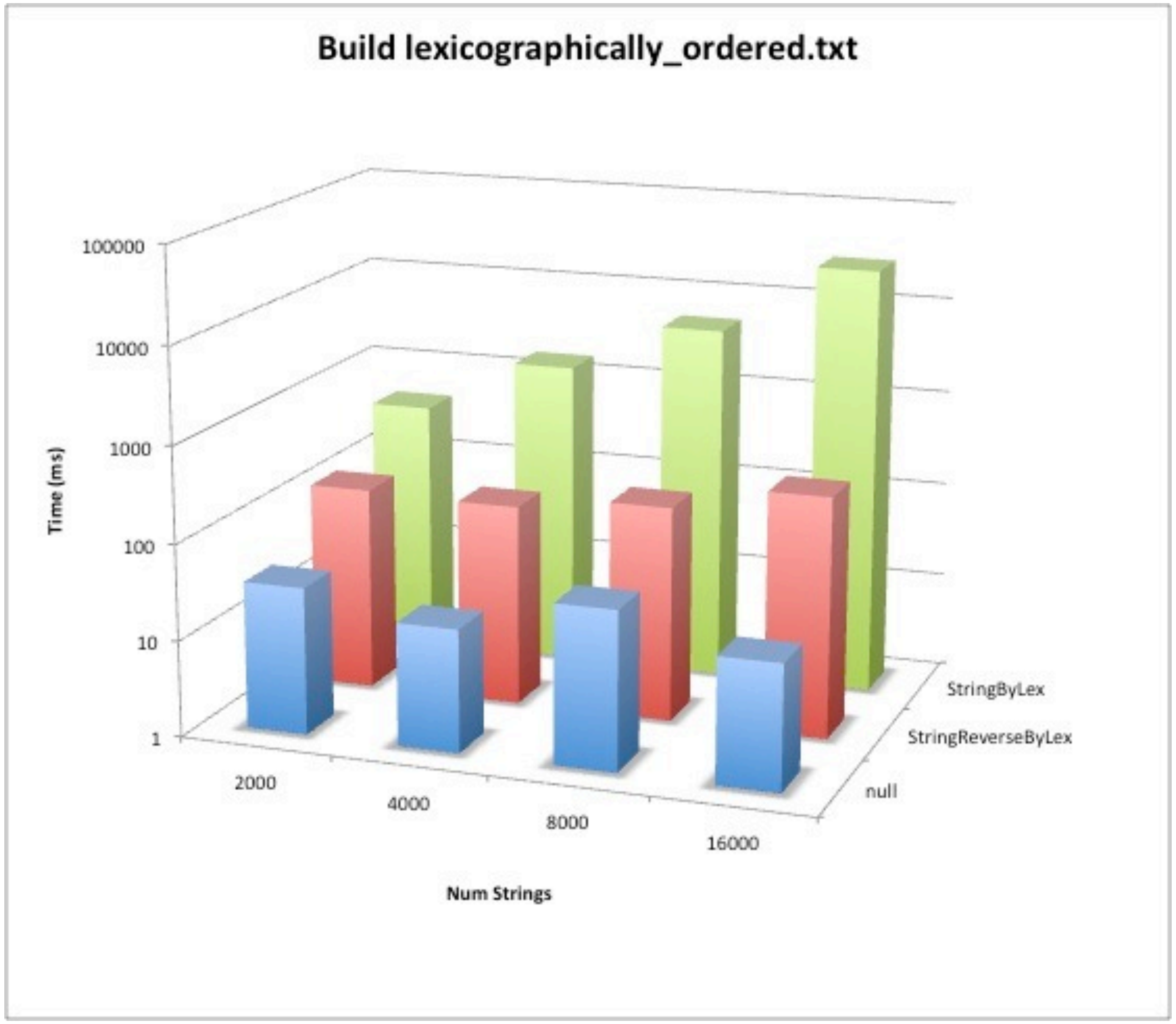
Assignment 5

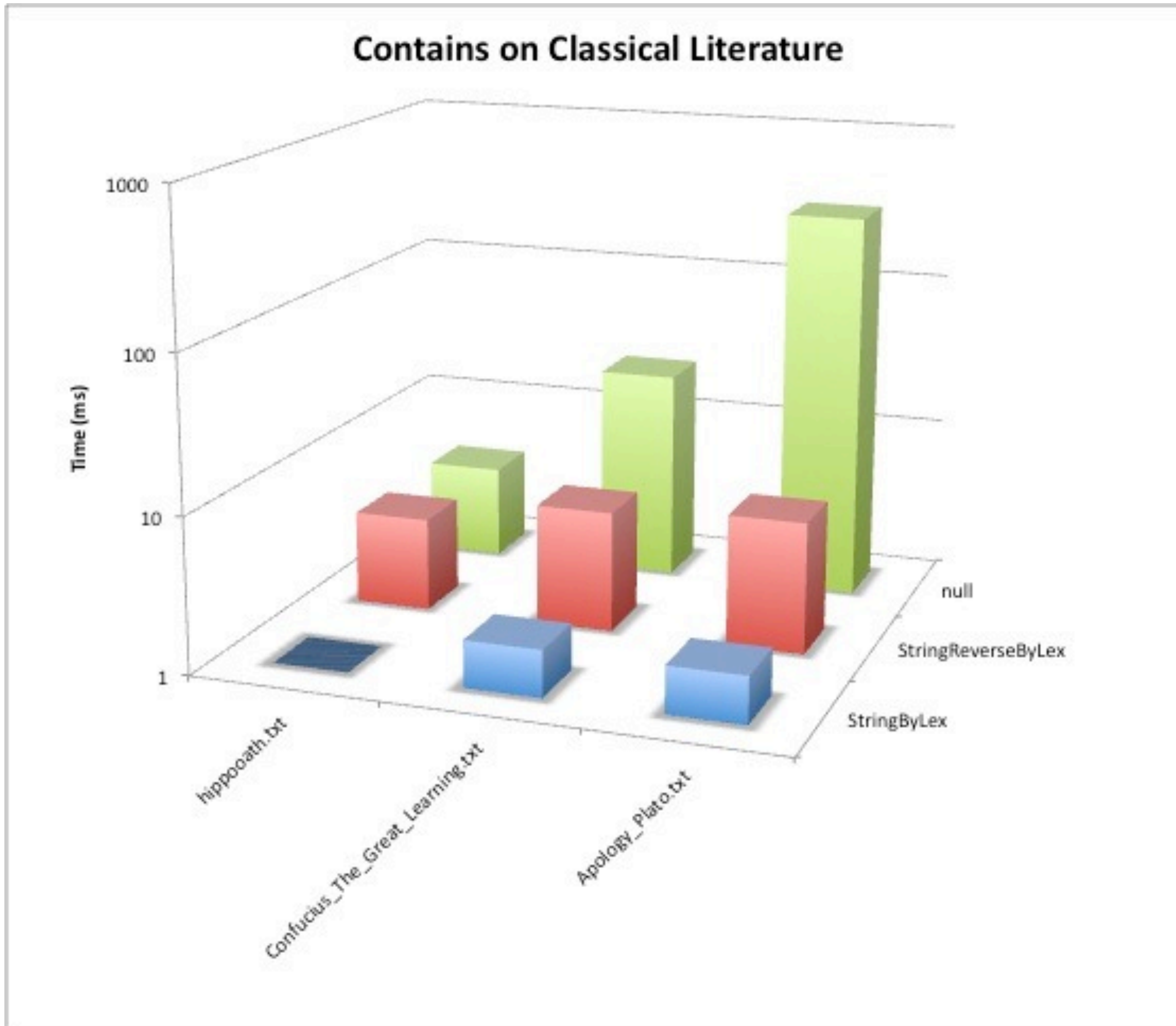
- Due Tuesday, February 11 at 11:59pm
- More efficient MyMap<K,V> using binary search tree

Assignment 6

- Timing program for MyMap<K,V>
- Due: Friday, February 14, 2014 at 11:59pm

	A	B	C	D	E	F	G	H	I
1	**Comparator**	**File**	**Num Strings**	**Size (#)**	**Build (ms)**	**Iterator (ms)**	**Iterate (ms)**	**Contains (ms)**	**Num Contained**
2	null	lexicographically_ordered.txt	2000	2000	34	80	2	45	77
3	null	lexicographically_ordered.txt	4000						
4	null	lexicographically_ordered.txt	8000						
5	null	lexicographically_ordered.txt	16000						
6	null	random_order.txt	2000						
7	null	random_order.txt	4000						
8	null	random_order.txt	8000						
9	null	random_order.txt	16000						
10	null	reverse_ordered.txt	2000						
11	null	reverse_ordered.txt	4000						
12	null	reverse_ordered.txt	8000						
13	null	reverse_ordered.txt	16000						
14	StringByLex	lexicographically_ordered.txt	2000						
15	StringByLex	lexicographically_ordered.txt	4000						
16	StringByLex	lexicographically_ordered.txt	8000						
17	StringByLex	lexicographically_ordered.txt	16000						
18	StringByLex	random_order.txt	2000						
19	StringByLex	random_order.txt	4000						
20	StringByLex	random_order.txt	8000						
21	StringByLex	random_order.txt	16000						
22	StringByLex	reverse_ordered.txt	2000						
23	StringByLex	reverse_ordered.txt	4000						
24	StringByLex	reverse_ordered.txt	8000						
25	StringByLex	reverse_ordered.txt	16000						
26	StringReverseByLex	lexicographically_ordered.txt	2000						
27	StringReverseByLex	lexicographically_ordered.txt	4000						
28	StringReverseByLex	lexicographically_ordered.txt	8000						
29	StringReverseByLex	lexicographically_ordered.txt	16000						
30	StringReverseByLex	random_order.txt	2000						
31	StringReverseByLex	random_order.txt	4000						
32	StringReverseByLex	random_order.txt	8000						
33	StringReverseByLex	random_order.txt	16000						
34	StringReverseByLex	reverse_ordered.txt	2000						
35	StringReverseByLex	reverse_ordered.txt	4000						
36	StringReverseByLex	reverse_ordered.txt	8000						
37	StringReverseByLex	reverse_ordered.txt	16000						


Build lexicographically_ordered.txt
100000
10000
1000
100
10
1
Time (ms)
2000
4000
8000
16000
Num Strings
StringByLex
StringReverseByLex
null


Contains on Classical Literature
Time (ms)
1000
100
10
1
hippooath.txt
Confucius_The_Great_Learning.txt
Apology_Plato.txt
null
StringReverseByLex
StringByLex

Total Order

A *total order* on some set D is a binary relation R on D such that

- R is transitive
- R is anti-symmetric
- R satisfies the law of trichotomy

Tree Basics
[Lewis & Chase]

- Tree: “a non-linear structure in which elements are organized into a hierarchy”
  - Tree contains nodes (elements) and edges (connect nodes)
- Root: single node at top level of tree
- “The nodes at lower levels of the tree are the children of nodes at the previous level. Nodes that have the same parent are called siblings.”
- Leaf: “node that does not have any children”
- Internal node: “node that is not the root and has at least one child”

Labeled Binary Tree (LBT)

- an empty tree
- a node with three components:
  - a label
  - a left subtree, which is a labeled binary tree
  - a right subtree, which is a labeled binary tree

Binary Search Tree (BST)

- t is empty
- t is a node
  - a label
  - the left subtree of t is a BST,
  - the right subtree of t is a BST,
  - every label within the left subtree of t is less than the label of t,
  - every label within the right subtree of t is greater than the label of t

BST Invariants

- No duplicates
- left is a BST
- right is a BST
- all elements in left BST are less than current
- all elements in right BST are more than current

Algebraic Specs for BST

Asymptotic Notation

Why do we care about asymptotic notation?

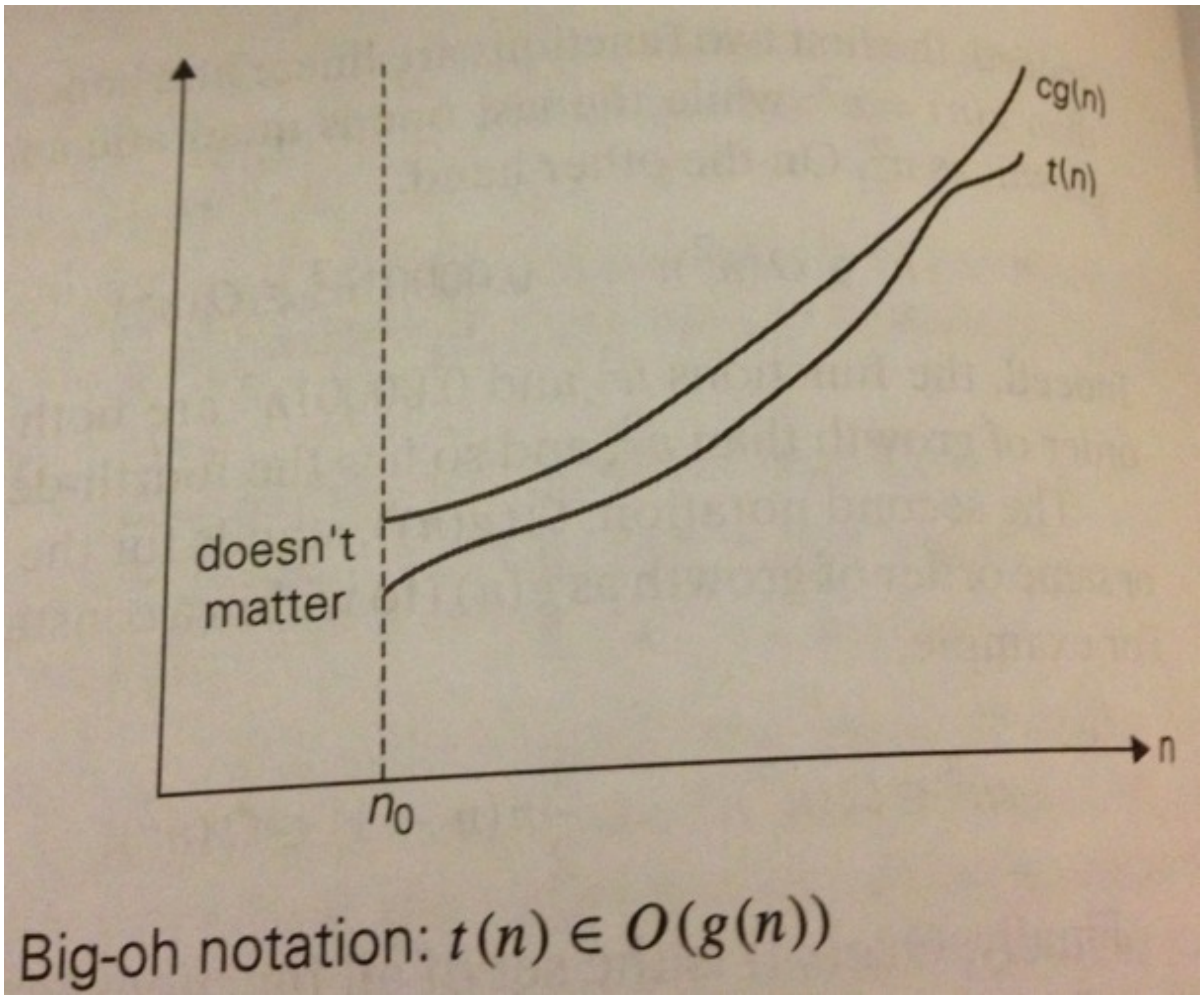
- Order of growth of a function
- Timing and efficiency of algorithms and implementations

Big-O

Big-O notation
[Levitin]

A function $t(x)$ is said to be in $O(g(x))$, denoted $t(x) \in O(g(x))$, if $t(x)$ is bounded above by some constant multiple of $g(x)$ for all large x, i.e., if there exist some positive constant c and some nonnegative integer $x0$ such that $t(x) \leq c * g(x)$ for all $x \geq x0$.

Big-O notation
[Levitin]



Big-O notation
[Clinger]

More general definition:

If $g : X \to R$, then
$O(g)=\{f : X \to R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0+c1 * g(x)\}$

Is $f(x) = x \in O(x^2)$?

Is $f(x) = x \in O(x^2)$?

We want to find c0 and c1 > 0 such that

$\forall x \in X, x <= c0 + c1 * x^2$

Northeastern University
College *of* Computer and Information Science

Is $f(x) = x \in O(x^2)$?

We want to find c0 and c1 > 0 such that

$\forall x \in X, x <= c0 + c1 * x^2$

What about c0 = 1 and c1 = 1?

Is $f(x) = x \in O(x^2)$?

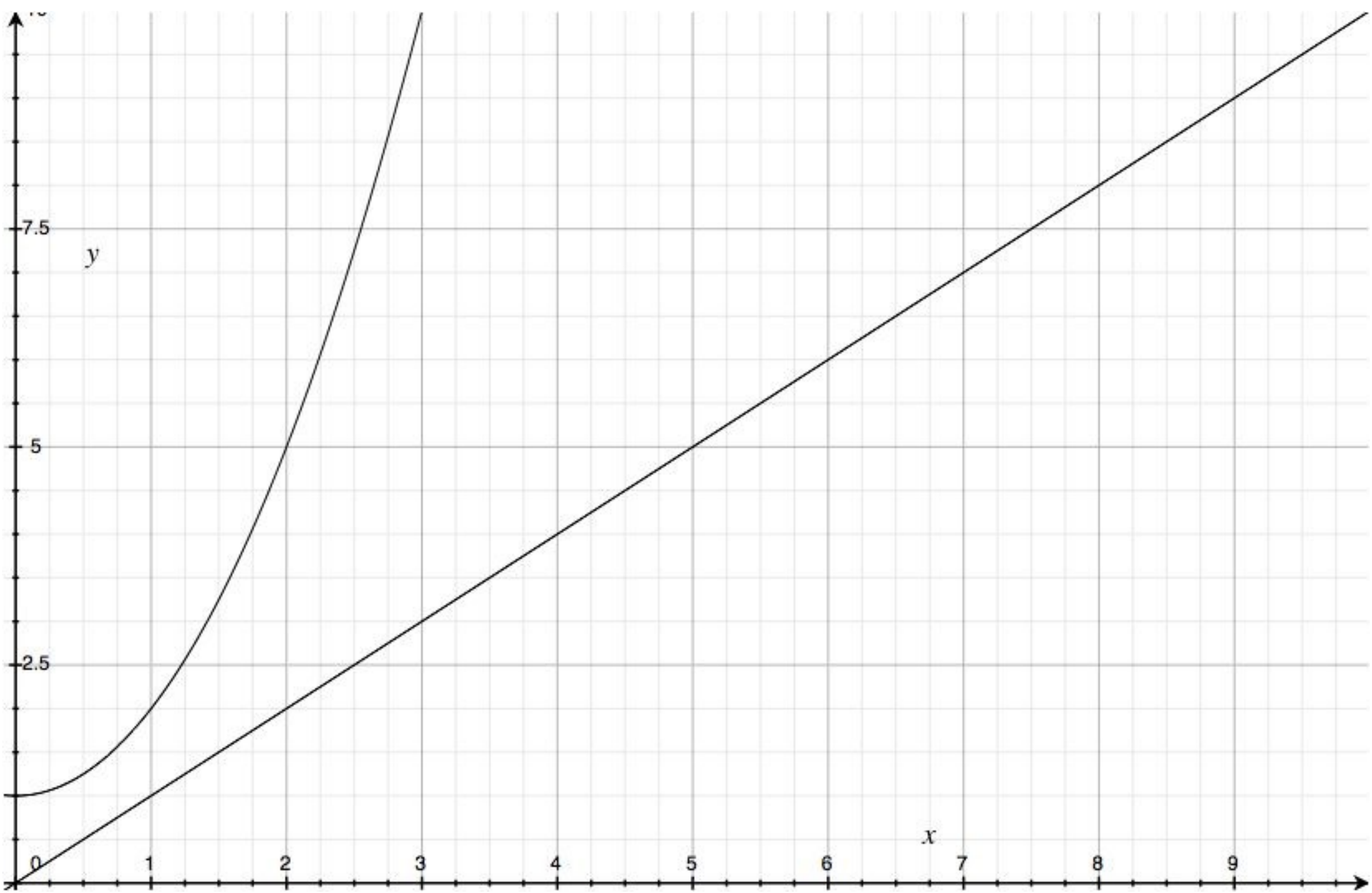
We want to find c0 and c1 > 0 such that

$\forall x \in X, x <= c0 + c1 * x^2$

What about c0 = 1 and c1 = 1?

$\forall x \in X, x <= 1 + 1 * x^2$

Is $f(x) = x \in O(x^2)$?



Is $f(x) = x \in O(x^2)$?

Is $f(x) = x \in O(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \leq c * g(x)$ for all $x \geq x0$

Is $f(x) = x \in O(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \leq c * g(x)$ for all $x \geq x0$

We want to find: $x \leq c * x^2$ for all $x \geq x0$

Is $f(x) = x \in O(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \leq c * g(x)$ for all $x \geq x0$

We want to find: $x \leq c * x^2$ for all $x \geq x0$

What about $c = 1$ and $x0 = 1$?

Is $f(x) = x \in O(x\^2)$?

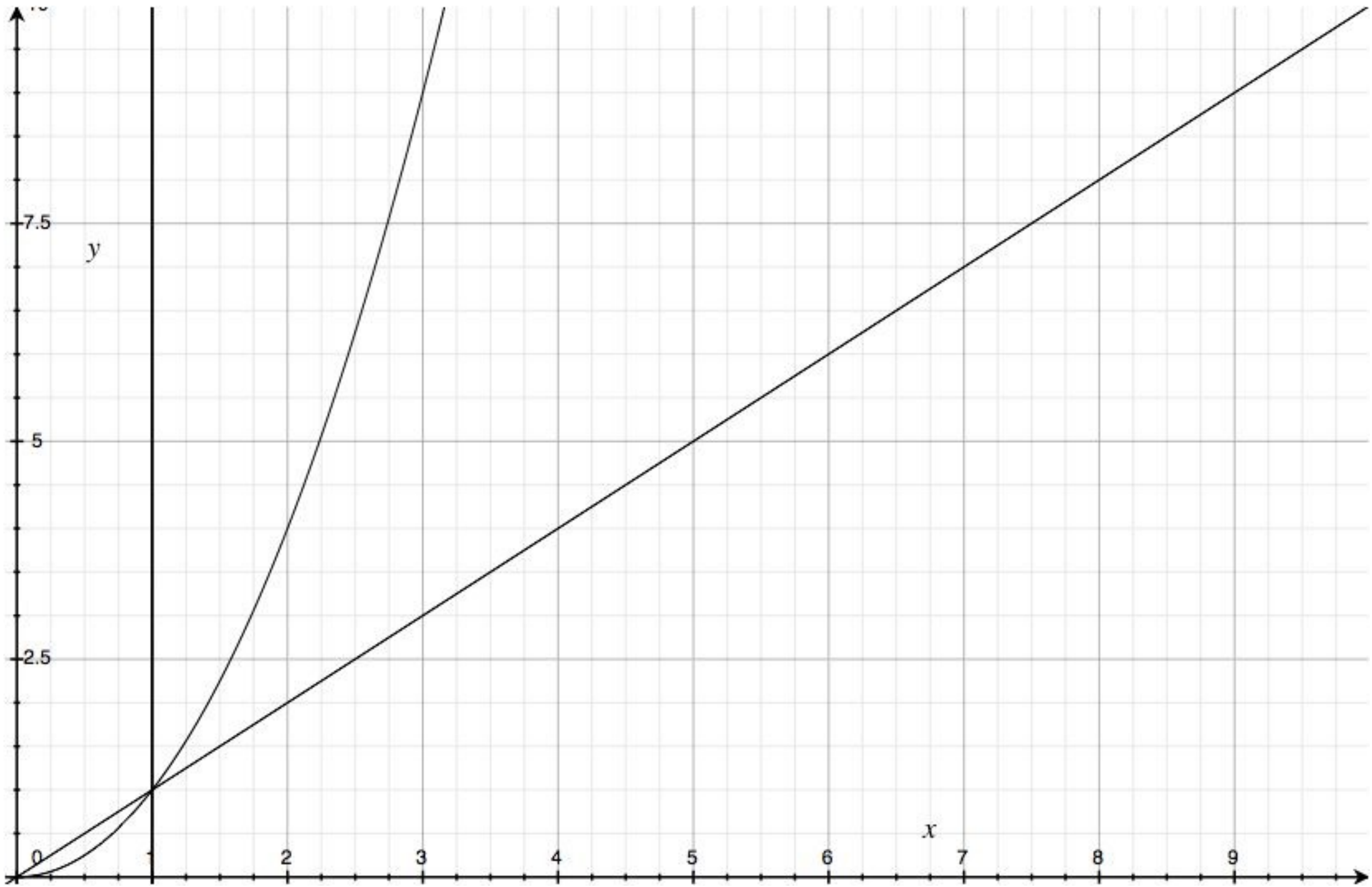
If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \leq c * g(x)$ for all $x \geq x0$

We want to find: $x \leq c * x\^2$ for all $x \geq x0$

What about $c = 1$ and $x0 = 1$?

$x \leq 1 * x\^2$ for all $x \geq 1$

Is f(x) = x ∈ O(x^2)?



Is $f(x) = 100x + 5 \in O(x^2)$?

Is $f(x) = 100x + 5 \in O(x^2)$?

$$O(g)=\{f : X \rightarrow R \mid \exists c0, c1 > 0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$$

Is $f(x) = 100x + 5 \in O(x\hat{}2)$?

$O(g)=\{f : X \to R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that $\forall x \in X$,
$100x + 5 <= c0 + c1 * x\hat{}2$.

Is $f(x) = 100x + 5 \in O(x^2)$?

$O(g)=\{f : X \rightarrow R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that $\forall x \in X$,
$100x + 5 <= c0 + c1 * x^2$.

What about c0 = 100 and c1 = 100?

Is $f(x) = 100x + 5 \in O(x^2)$?

$O(g)=\{f : X \rightarrow R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that $\forall x \in X$, $100x + 5 <= c0 + c1 * x^2$.

What about c0 = 100 and c1 = 100?

$\forall x \in X, 100x + 5 <= 100 + 100 * x^2$

Is $f(x) = 100x + 5 \in O(x^2)$?

Is $f(x) = 100x + 5 \in O(x)$?

Is $f(x) = 100x + 5 \in O(x)$?

$$O(g)=\{f : X \rightarrow R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$$

Northeastern University
College *of* Computer and Information Science

Is $f(x) = 100x + 5 \in O(x)$?

$O(g) = \{f : X \to R \mid \exists c0, c1 > 0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that
$\forall x \in X, 100x + 5 <= c0 + c1 * x.$

Is $f(x) = 100x + 5 \in O(x)$?

$O(g)=\{f : X \to R \mid \exists c0, c1>0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that
$\forall x \in X$, $100x + 5 <= c0 + c1 * x$.

What about c0 = 5 and c1 = 100?

Is $f(x) = 100x + 5 \in O(x)$?

$O(g) = \{f : X \to R \mid \exists c0, c1 > 0, \forall x \in X, f(x) <= c0 + c1 * g(x)\}$

We want to find c0 and c1 > 0 such that
$\forall x \in X, 100x + 5 <= c0 + c1 * x.$

What about c0 = 5 and c1 = 100?

$\forall x \in X, 100x + 5 <= 100x + 5$

Sum of Integers from 1 to n

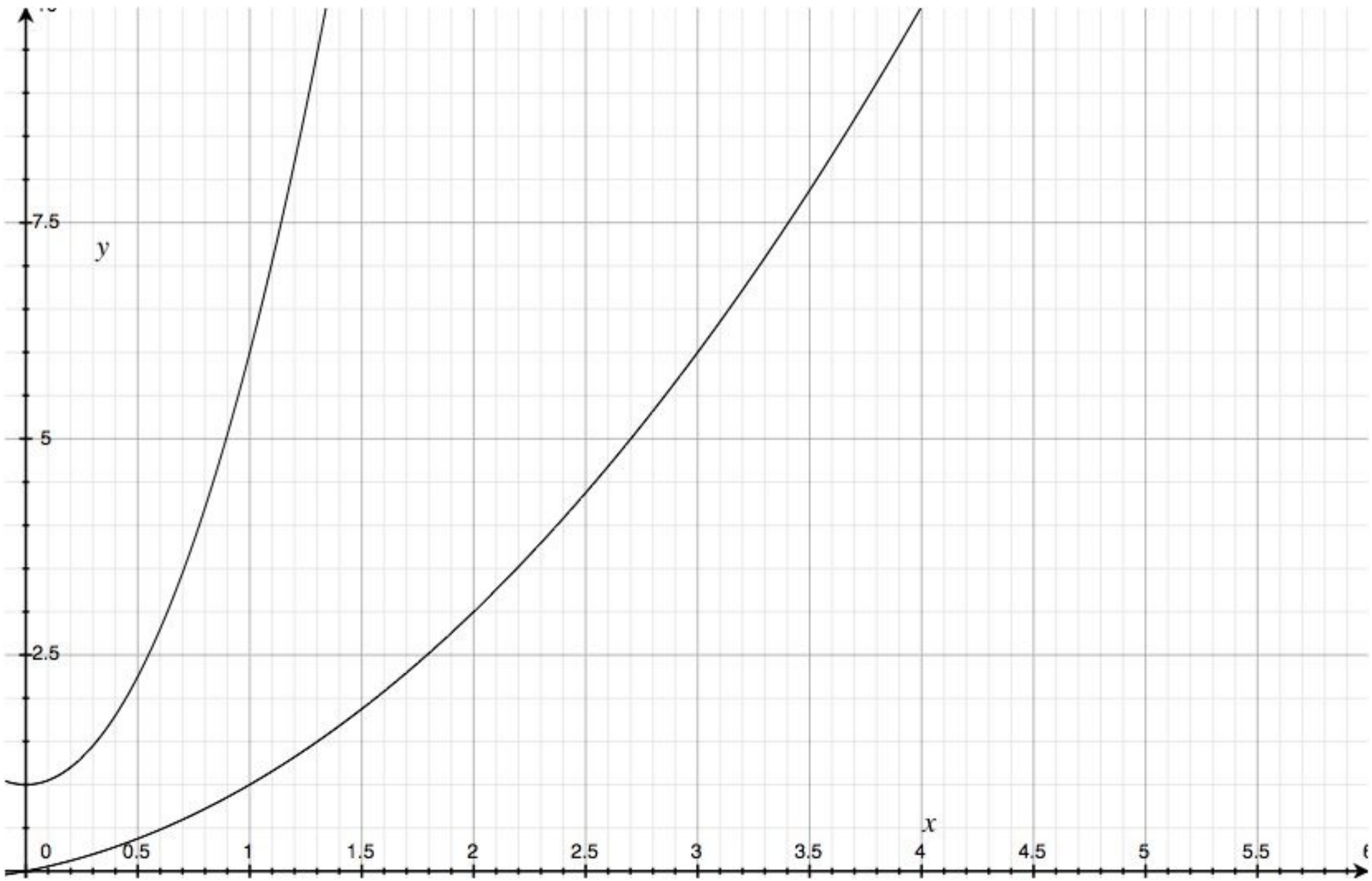
Sum of Integers from 1 to n

$$\sum_{1}^{n} i$$

Sum of Integers from 1 to n

$$\sum_{1}^{n} i \in O(n^2)$$

$\forall x \in X,\ (x^2+x)/2 <= 1+5*x^2$



Theorem

If f(x) is a polynomial function of x of degree k, with positive leading coefficient and restricted to non-negative x, then

$$O(f) = O(x^k).$$

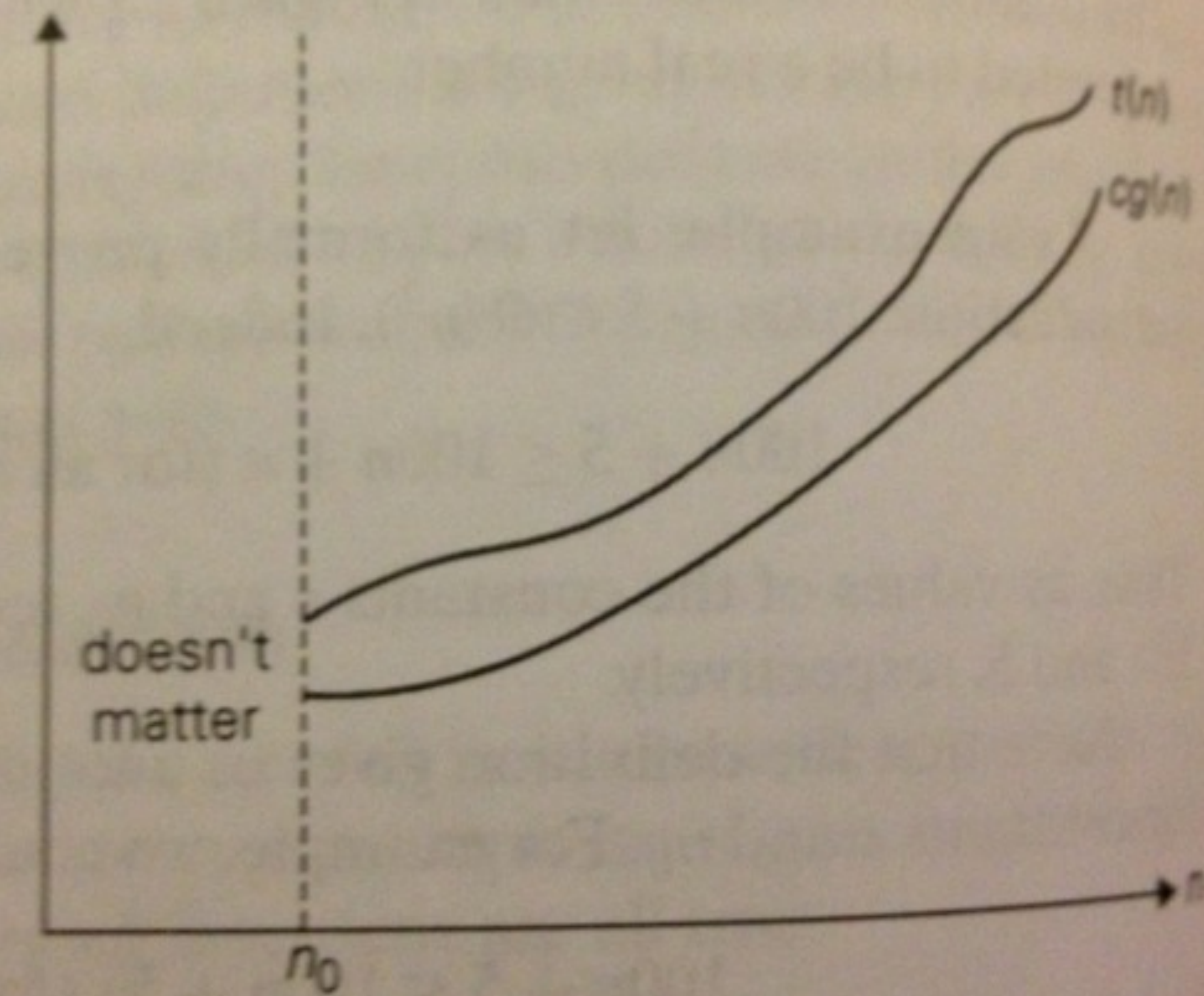
Big Omega

Big-Omega notation

[Levitin]

A function $t(x)$ is said to be in $\Omega(g(x))$, denoted $t(x) \in \Omega(g(x))$, if $t(x)$ is bounded below by some positive constant multiple of $g(x)$ for all large x, i.e., if there exist some positive constant c and some nonnegative integer x0 such that
$t(x) \geq c * g(x)$ for all $x \geq x0$.

Big-Omega notation
[Levitin]



Big-omega notation: $t(n) \in \Omega(g(n))$

Is $f(x) = x^3 \in \Omega(x^2)$?

Is $f(x) = x^3 \in \Omega(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \geq c * g(x)$ for all $x \geq x0$.

Is $f(x) = x^3 \in \Omega(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \geq c * g(x)$ for all $x \geq x0$.

Want $x^3 \geq c * x^2, \forall x \geq x0$.

Northeastern University
College *of* Computer and Information Science

Is $f(x) = x^3 \in \Omega(x^2)$?

If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \geq c * g(x)$ for all $x \geq x0$.

Want $x^3 \geq c * x^2, \forall x \geq x0$.

What about $c = 1$ and $x0 = 1$?

Is $f(x) = x^3 \in \Omega(x^2)$?

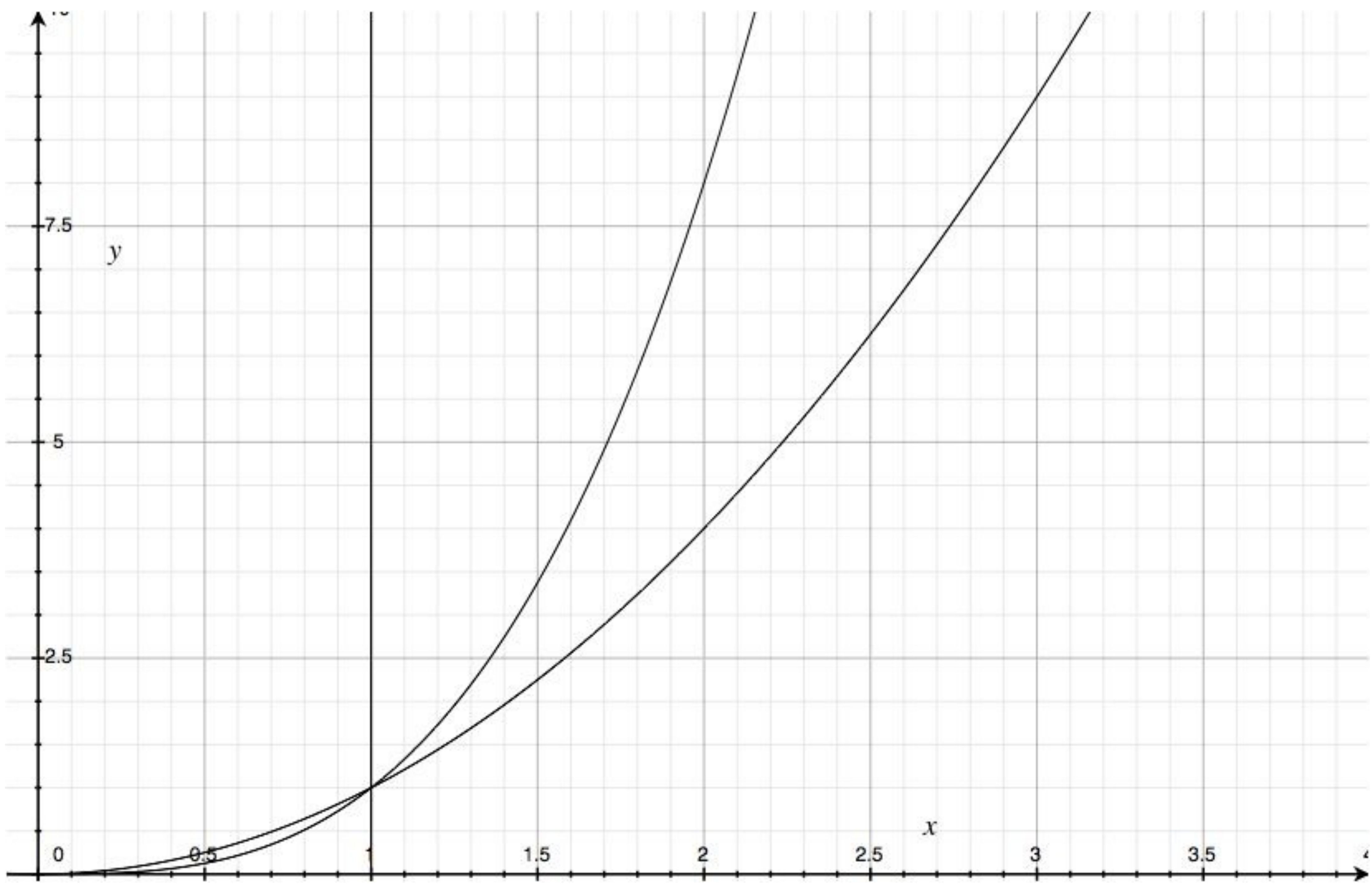
If there exist some positive constant c and some nonnegative integer x0 such that $t(x) \geq c * g(x)$ for all $x \geq x0$.

Want $x^3 \geq c * x^2, \forall x \geq x0$.

What about $c = 1$ and $x0 = 1$?

$$x^3 \geq 1 * x^2, \forall x \geq 1$$

Is $f(x) = x^3 \in \Omega(x^2)$?

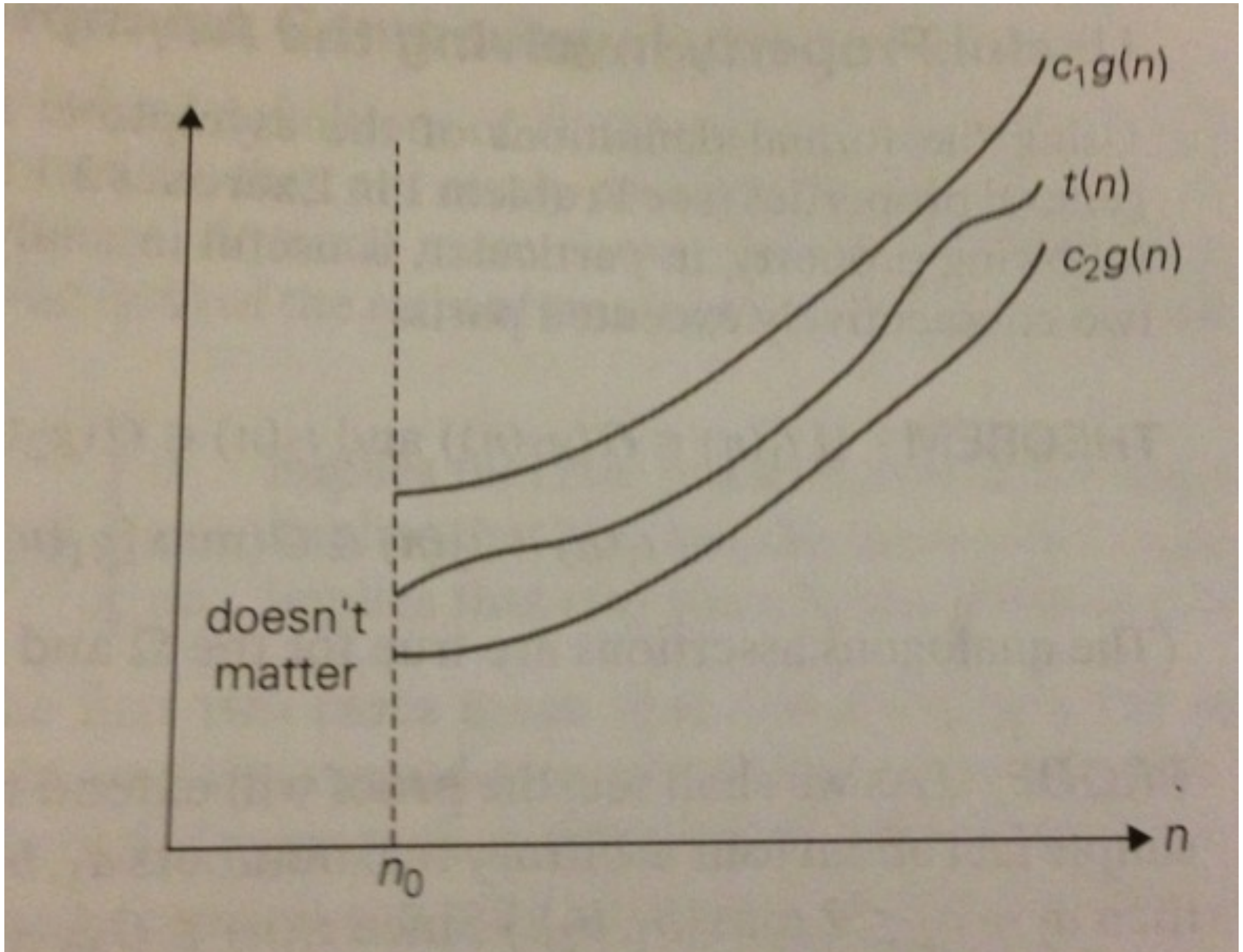


Big-Theta

Big-Theta notation
[Levitin]

A function $t(x)$ is said to be in $\Theta(g(x))$, denoted $t(x) \in \Theta(g(x))$, if $t(x)$ is bounded both above and below by some positive constant multiples of $g(x)$ for all large x, i.e., if there exist some positive constant c1 and c2 and some nonnegative integer x0 such that $c2 * g(x) \leq t(x) \leq c1 * g(x)$ for all $x \geq x0$.

Big-Theta notation
[Levitin]



Big-theta notation: $t(n) \in \Theta(g(n))$

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?

If there exist some positive constant c1 and c2 and some nonnegative integer x0 such that

$$c2 * g(x) \leq t(x) \leq c1 * g(x) \text{ for all } x \geq x0.$$

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?

If there exist some positive constant c1 and c2 and some nonnegative integer x0 such that

$$c2 * g(x) \leq t(x) \leq c1 * g(x) \text{ for all } x \geq x0.$$

Want $c2 * x^3 \leq x^3 + 4 * x^2 + 5 \leq c1 * x^3$,

$\forall x \geq x0$.

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?

If there exist some positive constant c1 and c2 and some nonnegative integer x0 such that

$c2 * g(x) \leq t(x) \leq c1 * g(x)$ for all $x \geq x0$.

Want $c2 * x^3 \leq x^3 + 4 * x^2 + 5 \leq c1 * x^3$,

$\forall x \geq x0$.

What about c1 = 10, c2 = 1 and x0 = 1?

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?

If there exist some positive constant c1 and c2 and some nonnegative integer x0 such that

$$c2 * g(x) \leq t(x) \leq c1 * g(x) \text{ for all } x \geq x0.$$

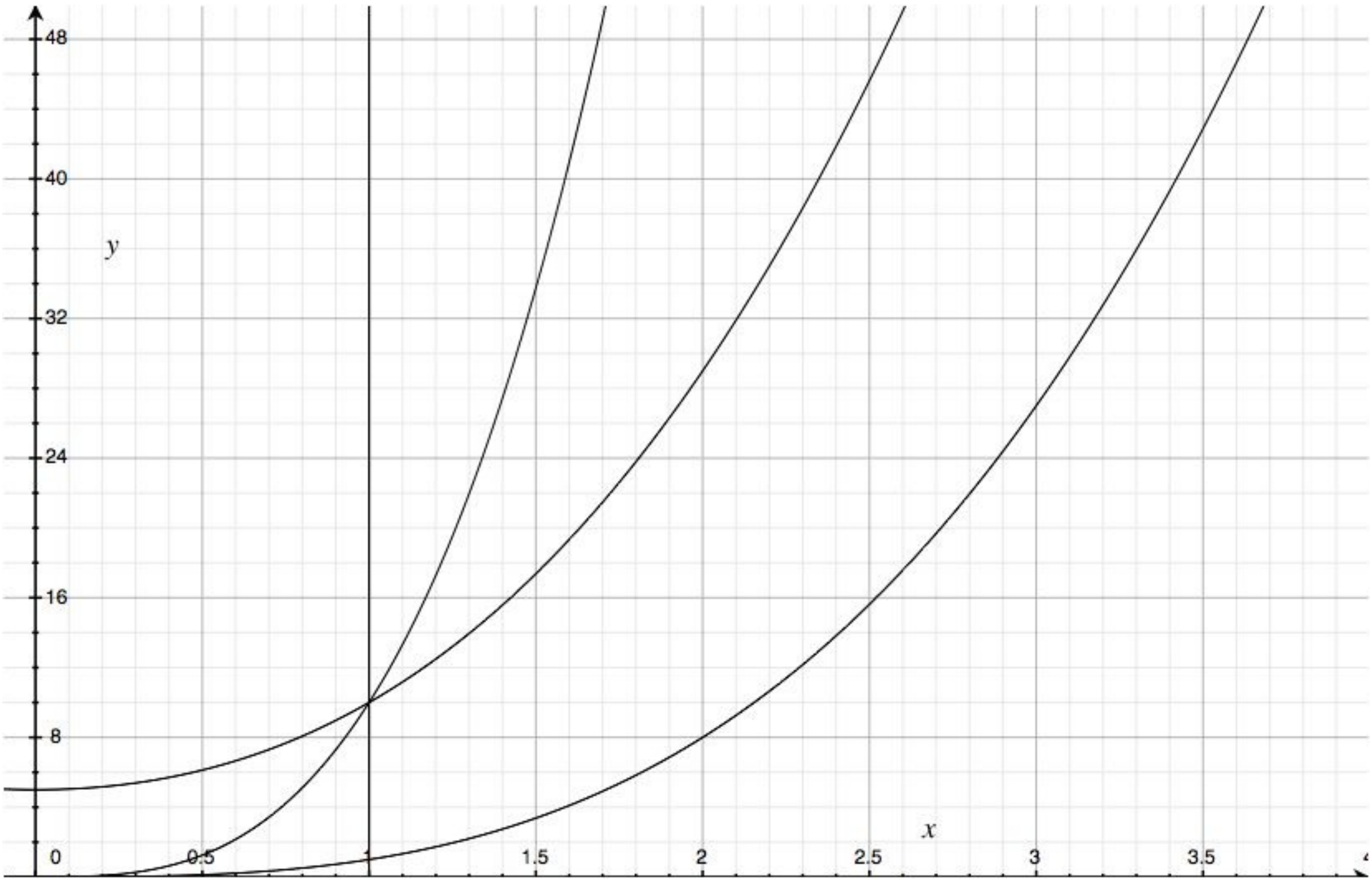
Want $c2 * x^3 \leq x^3 + 4 * x^2 + 5 \leq c1 * x^3$,

$\forall x \geq x0$.

What about c1 = 10, c2 = 1 and x0 = 1?

$$1 * x3 \leq x3 + 4 * x2 + 5 \leq 10 * x3, \forall x \geq 1$$

Is $f(x) = x^3 + 4 * x^2 + 5 \in \Theta(x^3)$?



Let's play a game...

Let's play a game...

$f(x) = x * (x + 1)/2$

Let's play a game...

$f(x) = x * (x + 1)/2$

Let's assume $f \in O(1)$, then (by the general definition of big-O) there exist constants c0 and c1 such that $\exists c0, c1 > 0, \forall x \in X, f(x) \leq c0 + c1 * 1 = c0 + c1$

Let's play a game...

$f(x) = x * (x + 1)/2$

Let's assume $f \in O(1)$, then (by the general definition of big-O) there exist constants c0 and c1 such that $\exists c0, c1 > 0, \forall x \in X, f(x) \leq c0 + c1 * 1 = c0 + c1$

Pick values for c0 and c1.

Efficiency

n	log_2 n	n	n log_2 n	n^2	n^3	2^n	n!
10	3.3	10	3.3 * 10	10^2	10^3	1024	3628800
10^2	6.6	10^2	6.6 * 10^2	10^4	10^6	1.26765E+30	9.33262E+157
10^3	10	10^3	10 * 10^3	10^6	10^9	1.07151E+301	
10^4	13	10^4	13 * 10^4	10^8	10^12		
10^5	17	10^5	17 * 10^5	10^10	10^15		
10^6	20	10^6	20 * 10^6	10^12	10^18		

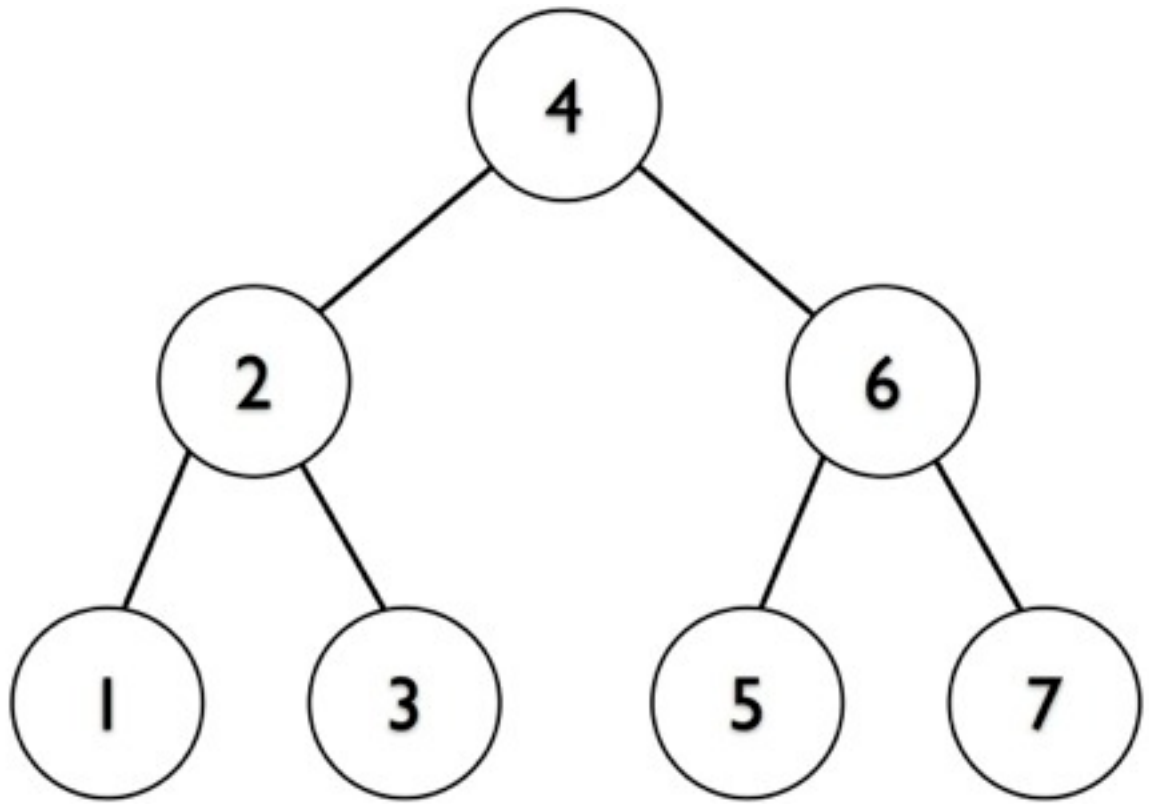
Efficiency

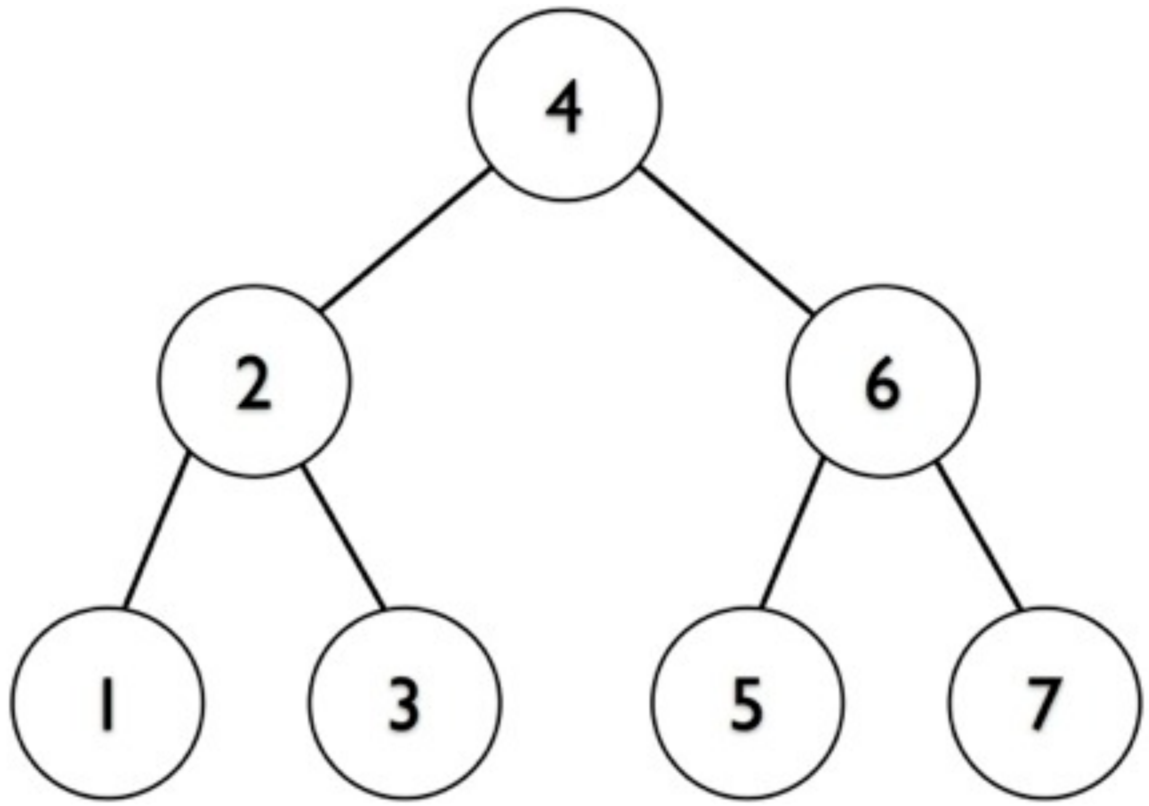
- Worst-case efficiency
- Best-case efficiency
- Average-case efficiency

Efficiency of Assignments with Recipe Implementations

- MySet
- MyList

Binary Search Tree Efficiency


4
2
6
1
3
5
7



$$(1 * 1 + 2 * 2 + 4 * 3)/7 = 17/7 \approx 2.42$$

h	n	sum	average search
1	1	1	1
2	3	5	1.66666667
3	7	17	2.42857143
4	15	49	3.26666667
5	31	129	4.16129032
6	63	321	5.0952381
7	127	769	6.05511811
8	255	1793	7.03137255
9	511	4097	8.01761252
10	1023	9217	9.00977517
11	2047	20481	10.0053737
12	4095	45057	11.0029304
13	8191	98305	12.0015871
14	16383	212993	13.0008545
15	32767	458753	14.0004578
16	65535	983041	15.0002441
17	131071	2097153	16.0001297
18	262143	4456449	17.0000687
19	524287	9437185	18.0000362
20	1048575	19922945	19.0000191
21	2097151	41943041	20.00001
22	4194303	88080385	21.0000052
23	8388607	184549377	22.0000027
24	16777215	385875969	23.0000014
25	33554431	805306369	24.0000007
26	67108863	1677721601	25.0000004
27	134217727	3489660929	26.0000002
28	268435455	7247757313	27.0000001

Optimization

"More computing sins are committed in the name of efficiency (without necessarily achieving it) than for any other single reason—including blind stupidity."
–W. A. Wulf

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil"

–Donald Ervin Knuth

Rules of Optimization

1. Don't.
2. Don't yet.
3. Don't optimize more than necessary.