04/04/2025: Big O Notation

CSCI 246: Discrete Structures

Textbook reference: Sec 7.2, Ponomarenko

Graded Quiz Pickup

Quizzes are in the front of the room, grouped into four bins (A-G, H-L, M-R, S-Z) by last name. The quizzes are upside down with your last name on the back. Come find yours before, during, or after class. Only turn the quiz over if it's yours.

Today's Agenda

- Reading and problems quizzes (15 mins)
- Mini-lecture ($\approx$ 15 mins)
- Group exercises ($\approx$ 15 mins)

Feedback on Wednesday’s Quiz

Reading Quiz Scores (Extra Credit)

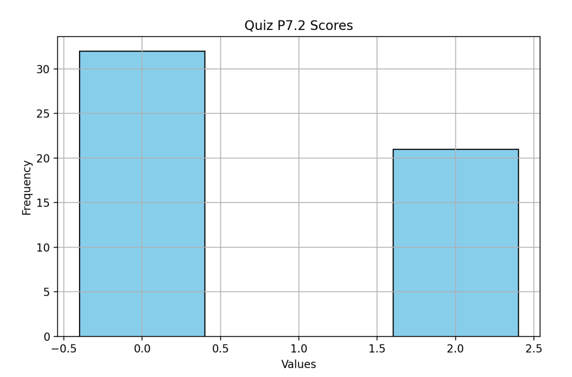


Figure 1: Median Score = 0 points extra credit

Grading Rubric: 2 points extra credit for perfect answer.

Today’s quiz

Problems Quiz (Conditional Prob, Random Variables, Expectation)

1. Let (S, P) be the sample space with $S = \{1, 2, \ldots, 10\}$ and $P(x) = \frac{1}{10}$ for all $x \in S$. Let A be the event "is even" and B be the event "is prime". Calculate $P(A \mid B)$ and $P(B \mid A)$. Show your work.
2. Two cards are drawn at random (without replacement) from a standard deck of 52 cards. Let X be the rank of the first card and Y be the rank of the second card. Are X and Y independent? Justify your answer with a mathematical argument.

Reading Quiz (Big O notation)

Consider the sequences $a_n = 3n + 100$ and $b_n = n$. Show that $a_n = O(b_n)$. That is, show that there is some $n_0 \in \mathbb{N}$ and some $M \in \mathbb{R}$ such that for every $n \geq n_0$, we have $|a_n| \leq M|b_n|$.

Thoughts On Big O Notation

How to compare the efficiency of algorithms?

Figure 2: German mathematician Paul Bachmann

How to compare the efficiency of algorithms?

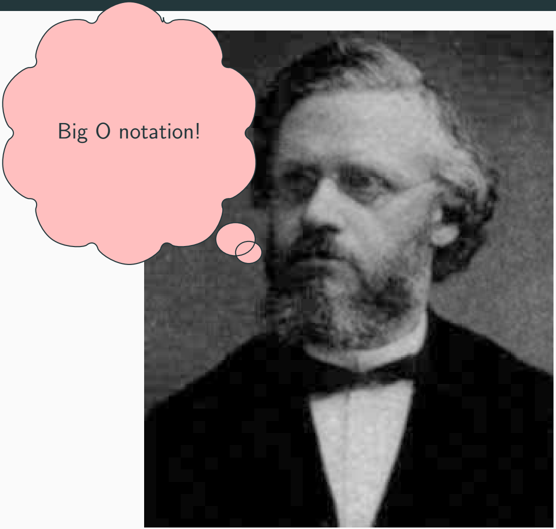


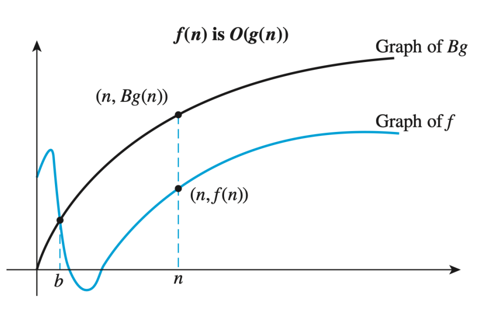
Figure 2: German mathematician Paul Bachmann

Definition

Let f and g be real-valued non-negative functions defined on the same set of nonnegative integers.

Then f is of order at most g, written $\boldsymbol{f(n)}$ **is** $\boldsymbol{O(g(n))}$ (f of n is big-O of g of n) if and only if there exist positive real number B and integer b such that

$$|f(n)| \leq B|g(n)| \quad \text{for every integer } n \geq b$$

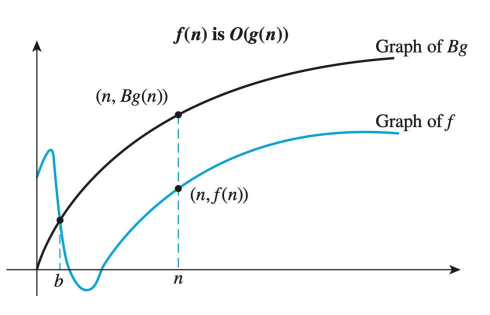


Definition

Let f and g be real-valued non-negative functions defined on the same set of nonnegative integers.

Then f is of order at most g, written $\boldsymbol{f(n)}$ **is** $\boldsymbol{O(g(n))}$ (f of n is big-O of g of n) if and only if there exist positive real number B and integer b such that

$$|f(n)| \leq B|g(n)| \quad \text{for every integer } n \geq b$$



Poll. How would you summarize this definition in words?

Poll. How would you summarize this definition in words?

Poll. How would you summarize this definition in words?

Solution. The values of f are eventually less than those of a positive multiple of g.

Poll. How would you summarize this definition in words?

Solution. The values of f are eventually less than those of a positive multiple of g.

We say that "f is of order at most g".

Poll. How would you summarize this definition in words?

Solution. The values of f are eventually less than those of a positive multiple of g.

We say that "f is of order at most g".

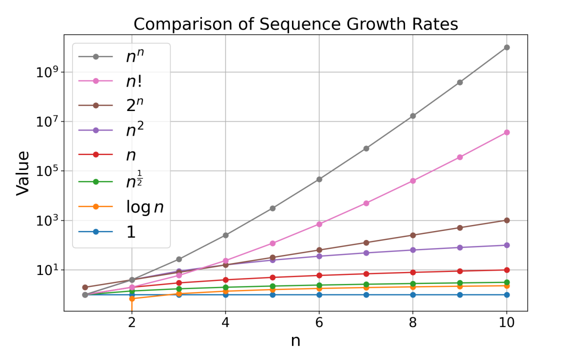
Poll. How would you summarize this definition in words?

Solution. The values of f are eventually less than those of a positive multiple of g.

We say that "f is of order at most g".

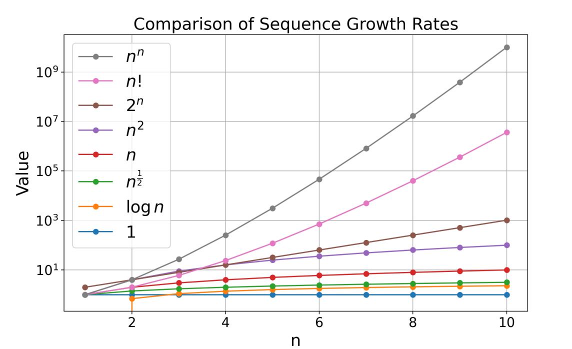
Poll. What's the deal with the highlighted words (especially positive multiple)?

Classification Theorem



Poll. How should we interpret this graph?

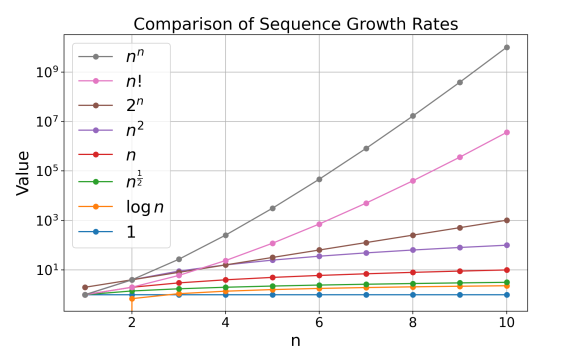
Classification Theorem



Poll. How should we interpret this graph?

- A taxonomy of growth rates (or scaling behavior): Any function is big O of everything equal to or above it, but is **not** big O of anything below it.

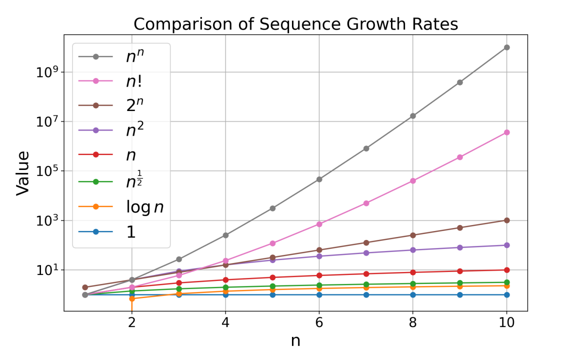
Classification Theorem



Poll. How should we interpret this graph?

- A taxonomy of growth rates (or scaling behavior): Any function is big O of everything equal to or above it, but is **not** big O of anything below it.
- **Example**: n is big O of n, n^2, 2^n, $n!$, and n^n. Even if we multiply n by a large constant, those functions will always eventually get bigger!

Classification Theorem



Poll. How should we interpret this graph?

- A taxonomy of growth rates (or scaling behavior): Any function is big O of everything equal to or above it, but is **not** big O of anything below it.
- **Example**: n is big O of n, n^2, 2^n, $n!$, and n^n. Even if we multiply n by a large constant, those functions will always eventually get bigger!
- **Anti-Example**: n is not big O of $n^{\frac{1}{2}}$. No matter how much you scale up $n^{\frac{1}{2}}$ by a large constant, n will eventually get bigger!

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

Example: $f(n) = 9n^2 + 3n - 2$ is $O(n^2)$.

Proof. Take $n \geq 1$. Then

$$|9n^2 + 3n - 2| \leq |9n^2 + 3n^2 + 2n^2| = 14|n^2|.$$

(To satisfy the definition of big O, we take $b = 1, B = 14$.)

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

Example: $f(n) = 9n^2 + 3n - 2$ is $O(n^2)$.

Proof. Take $n \geq 1$. Then

$$|9n^2 + 3n - 2| \leq |9n^2 + 3n^2 + 2n^2| = 14|n^2|.$$

(To satisfy the definition of big O, we take $b = 1, B = 14$.)

Remark. Big O notation hides irrelevant details.

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

Example: $f(n) = 9n^2 + 3n - 2$ is $O(n^2)$.

Proof. Take $n \geq 1$. Then

$$|9n^2 + 3n - 2| \leq |9n^2 + 3n^2 + 2n^2| = 14|n^2|.$$

(To satisfy the definition of big O, we take $b = 1, B = 14$.)

Remark. Big O notation hides irrelevant details.

Poll. Do you see a pattern?

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

Example: $f(n) = 9n^2 + 3n - 2$ is $O(n^2)$.

Proof. Take $n \geq 1$. Then

$$|9n^2 + 3n - 2| \leq |9n^2 + 3n^2 + 2n^2| = 14|n^2|.$$

(To satisfy the definition of big O, we take $b = 1, B = 14$.)

Remark. Big O notation hides irrelevant details.

Poll. Do you see a pattern?

Solution. Polynomials are big O of their highest-order term.

How to check Big O membership?

Example: $f(n) = 3n + 100$ is $O(n)$

Proof. Take $n \geq 1$. Then

$$|3n + 100| \leq |3n + 100n| = 103|n|.$$

(To satisfy the definition of big O, we take $b = 1, B = 103$.)

Example: $f(n) = 9n^2 + 3n - 2$ is $O(n^2)$.

Proof. Take $n \geq 1$. Then

$$|9n^2 + 3n - 2| \leq |9n^2 + 3n^2 + 2n^2| = 14|n^2|.$$

(To satisfy the definition of big O, we take $b = 1, B = 14$.)

Remark. Big O notation hides irrelevant details.

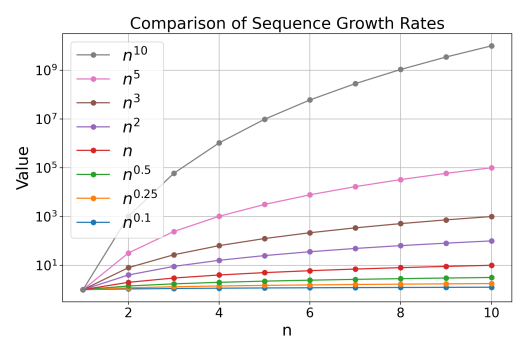
Poll. Do you see a pattern?

Solution. Polynomials are big O of their highest-order term.

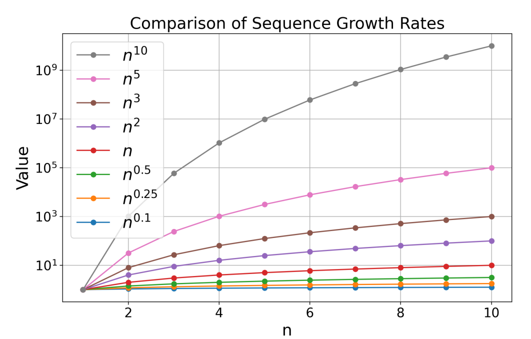
Remark. ...and also big O of still higher-order terms. E.g. for $n \geq 1$,

$$14|n^2| \leq 14|n^3| \leq 14|n^4| \leq \dots$$

Classification Theorem: Zooming In On Polynomials

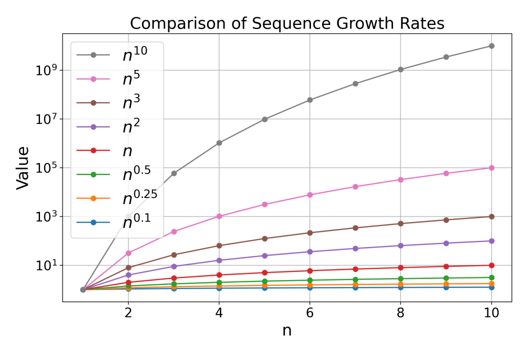


Classification Theorem: Zooming In On Polynomials



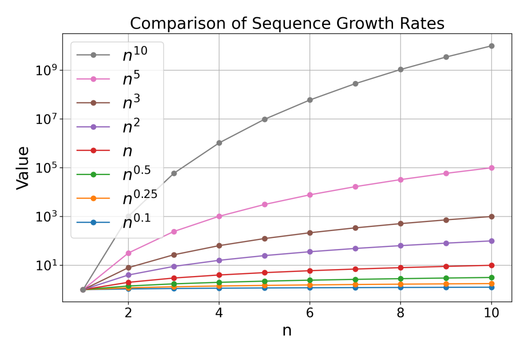
A taxonomy of growth rates: As you go down the legend, each function is big O of everything above it, but **not** big O of everything below it.

Classification Theorem: Zooming In On Polynomials



A taxonomy of growth rates: As you go down the legend, each function is big O of everything above it, but **not** big O of everything below it.

Classification Theorem: Zooming In On Polynomials



A taxonomy of growth rates: As you go down the legend, each function is big O of everything above it, but **not** big O of everything below it.

Theorem

For any function f and positive real numbers u and v with $u < v$,

$$f \text{ is } O(x^u) \implies f \text{ is } O(x^v)$$

A major problem with Big O statements

Poll. Suppose a person is analyzing the efficiency of algorithms, and finds that f is $O(x^5)$ and g is $O(x^4)$. Because $4 < 5$, this person concludes that g has a better algorithmic efficiency than f. Is this person's conclusion correct?

A major problem with Big O statements

Poll. Suppose a person is analyzing the efficiency of algorithms, and finds that f is $O(x^5)$ and g is $O(x^4)$. Because $4 < 5$, this person concludes that g has a better algorithmic efficiency than f. Is this person's conclusion correct?

Solution. No. For instance, take $f(x) = x^2$ and $g(x) = x^3$. Then

f is $O(x^2), O(x^3), O(x^4), O(x^5), \ldots$

and

g is $O(x^3), O(x^4), O(x^5), \ldots$

A major problem with Big O statements

Poll. Suppose a person is analyzing the efficiency of algorithms, and finds that f is $O(x^5)$ and g is $O(x^4)$. Because $4 < 5$, this person concludes that g has a better algorithmic efficiency than f. Is this person's conclusion correct?

Solution. No. For instance, take $f(x) = x^2$ and $g(x) = x^3$. Then

$$\text{f} \quad \text{is} \quad O(x^2), O(x^3), O(x^4), O(x^5), \ldots$$

and

$$\text{g} \quad \text{is} \quad O(x^3), O(x^4), O(x^5), \ldots$$

A major problem with Big O statements

Poll. Suppose a person is analyzing the efficiency of algorithms, and finds that f is $O(x^5)$ and g is $O(x^4)$. Because $4 < 5$, this person concludes that g has a better algorithmic efficiency than f. Is this person's conclusion correct?

Solution. No. For instance, take $f(x) = x^2$ and $g(x) = x^3$. Then

$$\text{f} \quad \text{is} \quad O(x^2), O(x^3), O(x^4), O(x^5), \ldots$$

and

$$\text{g} \quad \text{is} \quad O(x^3), O(x^4), O(x^5), \ldots$$

Remark. The problem is that the upper bounds for Big O can be needlessly large!

A major problem with Big O statements

Poll. Suppose a person is analyzing the efficiency of algorithms, and finds that f is $O(x^5)$ and g is $O(x^4)$. Because $4 < 5$, this person concludes that g has a better algorithmic efficiency than f. Is this person's conclusion correct?

Solution. No. For instance, take $f(x) = x^2$ and $g(x) = x^3$. Then

$$\text{f} \quad \text{is} \quad O(x^2), O(x^3), O(x^4), O(x^5), \ldots$$

and

$$\text{g} \quad \text{is} \quad O(x^3), O(x^4), O(x^5), \ldots$$

Remark. The problem is that the upper bounds for Big O can be needlessly large!

We should interpret "g is $O(x^4)$" as: "g is of order **at most** x^4".

How to compare the efficiency of algorithms?

Figure 3: American computer scientist and mathematician Donald Knuth

How to cor the efficiency of algorithms?

Big Theta notation!

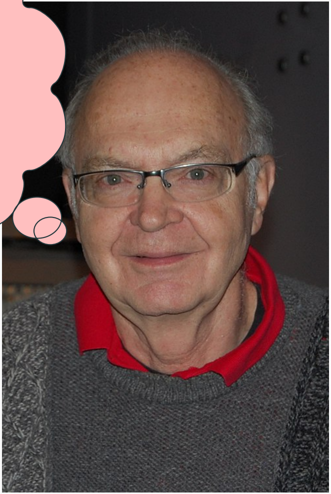
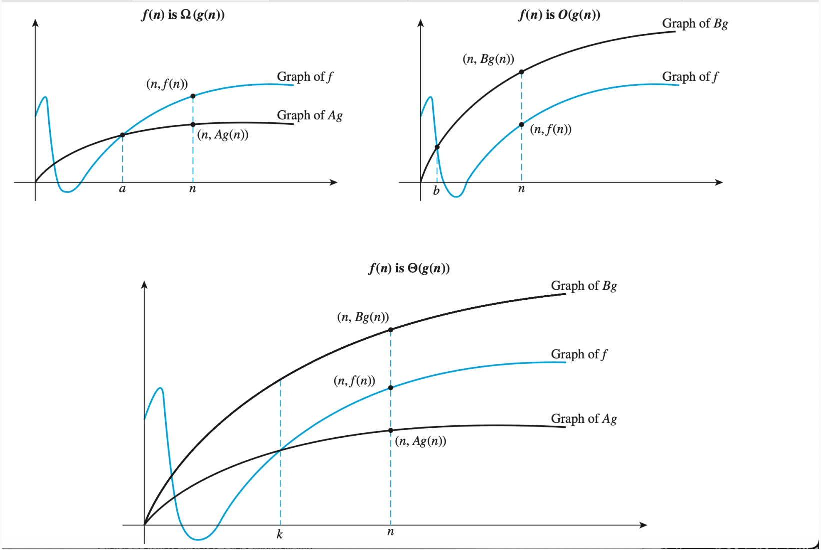


Figure 3: American computer scientist and mathematician Donald Knuth

Overview: Big Theta combines Big O and Big Omega

Big O	f is $O(g)$	f is of order at most g	The values of f are eventually **less** than those of a positive multiple of g.
Big Omega	f is $\Omega(g)$	f is of order at least g	The values of f are eventually **greater** than those of a positive multiple of g.
Big Theta	f is $\Theta(g)$	f is of order g	The values of f are eventually **less** than those of a positive multiple of g and **greater** than those of a positive multiple of g.

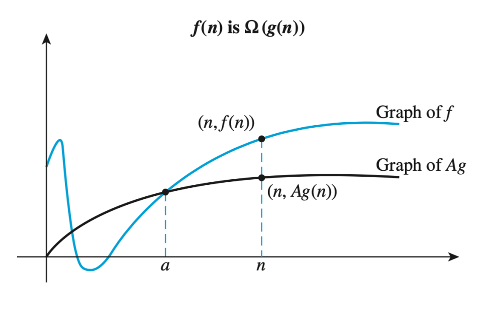

f(n) is Ω(g(n))
Graph of f
(n, f(n))
Graph of Ag
(n, Ag(n))
a
n
f(n) is O(g(n))
Graph of Bg
(n, Bg(n))
Graph of f
(n, f(n))
b
n
f(n) is Θ(g(n))
Graph of Bg
(n, Bg(n))
Graph of f
(n, f(n))
Graph of Ag
(n, Ag(n))
k
n

Big Omega: Definition

Definition

Let f and g be real-valued non-negative functions defined on the same set of nonnegative integers.
Then f is of order at least g, written $\boldsymbol{f(n)}$ **is** $\boldsymbol{\Omega(g(n))}$ (f of n is big-Omega of g of n) if and only if there exist positive real number A and integer a such that

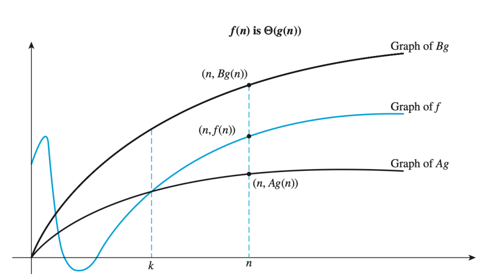
$$|f(n)| \geq A|g(n)| \quad \text{for every integer } n \geq a$$



Big Theta: Definition

Definition

Let f and g be real-valued non-negative functions defined on the same set of nonnegative integers.
Then f is of order g, written $\boldsymbol{f(n)}$ **is** $\boldsymbol{\Theta(g(n))}$ (f of n is big-Theta of g of n) if and only if $\boldsymbol{f(n)}$ is $\boldsymbol{O(g(n))}$ and $\boldsymbol{f(n)}$ is $\boldsymbol{\Omega(g(n))}$.



A useful theorem on Big Theta

Theorem: On Polynomial Orders

If m is any integer with $m \geq 0$ and $c_0, c_1, c_2, \ldots c_m$ are real numbers with $c_m \geq 0$, then

$$c_m x^m + c_{m-1} x^{m-1} + \ldots + c_1 x + c_0 \text{ is } \Theta(x^m).$$

Group exercises

aaron.loomis: 15
adam.wyszynski: 13
alexander.goetz: 2
alexander.knutson: 3
anthony.mann: 17
blake.leone: 19
bridger.voss: 19
caitlin.hermanson: 15
cameron.wittrock: 4
carsten.brooks: 7
carver.wambold: 8
colter.huber: 16
conner.reed1: 12
connor.mizner: 6
connor.yetter: 21
derek.price4: 14
devon.maurer: 2
emmeri.grooms: 4
erik.moore3: 3
ethan.johnson18: 8
evan.barth: 1
evan.schoening: 9
griffin.short: 14
jack.fry: 10
jacob.ketola: 11
jacob.ruiz1: 17
jacob.shepherd1: 4
jada.zorn: 13
jakob.kominsky: 12
james.brubaker: 10
jeremiah.mackey: 18
jett.girard: 21
john.fotheringham: 5
jonas.zeiler: 1
joseph.mergenthaler: 9
joseph.triem: 12
julia.larsen: 10
justice.mosso: 16
kaden.price: 13
lucas.jones6: 15
luka.derry: 17
luke.donaldson1: 7
lynsey.read: 14
mason.barnocky: 11
matthew.nagel: 1
micaylyn.parker: 6
michael.oswald: 20
nolan.scott1: 19
owen.obrien: 11
pendleton.johnston: 2
peter.buckley1: 6
reid.pickert: 18
ryan.barrett2: 3
samuel.hemmen: 20
samuel.mosier: 9
samuel.rollins: 20
sarah.periolat: 8
timothy.true: 16
tristan.nogacki: 5
tyler.broesel: 18
william.elder1: 21
yebin.wallace: 7
zeke.baumann: 5

Group exercises

Exercises.

1. Let $a_n = 1,000,000n + 3,000,000$. Prove that $a_n = O(n)$.
2. Let $a_n = 5 + \frac{1}{n} + \frac{1}{n+1}$. Prove that $a_n = O(1)$.
3. Let $a_n = n^2 + n + 1 + \frac{1}{n} + \sin n$. Prove that $a_n = O(n^2)$.
4. Let $a_n = 3n^2 + 7$. Prove that $a_n = \Theta(n^2)$.
5. Prove Theorem 7.9 from the Ponomarenko reading.

Theorem 7.9 (Ponomarenko). Let a_n be a sequence, and b_n, c_n be test sequences. If $a_n = O(b_n)$ and $b_n = O(c_n)$, then $a_n = O(c_n)$.

Solution to Group Exercise #1

Problem. Let $a_n = 1,000,000n + 3,000,000$. Prove that $a_n = O(n)$.

Solution. For $n \geq 1$,

$$|1,000,000n + 3,000,000| \leq |1,000,000n + 3,000,000n| = 4,000,000|n|$$

This satisfies the definition that a_n is $O(n)$ by setting $b = 1$ and $B = 4,000,000$.

Solution to Group Exercise #2

Problem. Let $a_n = 5 + \frac{1}{n} + \frac{1}{n+1}$. Prove that $a_n = O(1)$.

Solution. For $n \geq 1$,

$$|5 + \frac{1}{n} + \frac{1}{n+1}| \leq |5 + 1 + 1| = 7 \cdot |1|$$

This satisfies the definition that a_n is $O(1)$ by setting $b = 1$ and $B = 7$.

Solution to Group Exercise #3

Problem. Let $a_n = n^2 + n + 1 + \frac{1}{n} + \sin n$. Prove that $a_n = O(n^2)$.

Solution. For $n \geq 1$,

$$\begin{aligned} |n^2 + n + 1 + \frac{1}{n} + \sin n| &\leq |n^2 + n + 1 + 1 + 1| \\ &\leq |n^2 + n^2 + n^2 + n^2 + n^2| \\ &= 5|n^2| \end{aligned}$$

This satisfies the definition that a_n is $O(n^2)$ by setting $b = 1$ and $B = 5$.

Solution to Group Exercise #4

Problem. Let $a_n = 3n^2 + 7$. Prove that $a_n = \Theta(n^2)$.

Solution. For $n \geq 1$,

$$|3n^2 + 7| \leq |3n^2 + 7n^2| = 10|n^2|$$

This satisfies the definition that a_n is $O(n)$ by setting $b = 1$ and $B = 10$.

Again for $n \geq 1$,

$$|3n^2 + 7| \geq |3n^2| = 3|n^2|$$

This satisfies the definition that a_n is $\Omega(n^2)$ by setting $a = 1$ and $A = 3$.

Since a_n is both $O(n^2)$ and $\Omega(n^2)$, we can conclude that a_n is $\Theta(n^2)$.

Solution to Group Exercise #5

Problem. Prove Theorem 7.9 from the Ponomarenko reading.

Theorem 7.9 (Ponomarenko). Let a_n be a sequence, and b_n, c_n be test sequences. If $a_n = O(b_n)$ and $b_n = O(c_n)$, then $a_n = O(c_n)$.

Solution. Since $a_n = O(b_n)$, we have that for $n \geq b_1$,

$$|a_n| \leq B_1|b_n| \qquad \text{for some constant } \; B_1. \tag{1}$$

Since $b_n = O(c_n)$, we have that for $n \geq b_2$,

$$|b_n| \leq B_2|c_n| \qquad \text{for some constant } \; B_2. \tag{2}$$

Hence for $n \geq \mathtt{max}\{b_1, b_2\}$,

$$|a_n| \overset{\textit{Eq. }(1)}{\leq} B_1|b_n| \overset{\textit{Eq. }(2)}{\leq} B_1 \cdot B_2|c_n|$$

This satisfies the definition that a_n is $O(c_n)$ by setting $b = \mathtt{max}\{b_1, b_2\}$ and $B = B_1 \cdot B_2$.