COMP 250

Lecture 37

big Theta Θ
best and worst cases
limit rules

April 8, 2022

Previous two lectures

- big O – asymptotic upper bounds
- big Omega (Ω) – asymptotic lower bounds

Definition of Big Theta (Θ)

Let $t(n)$ and $g(n)$ be two functions of $n \geq 0$.

We say $t(n)$ is $\Theta(\, g(n)\,)$

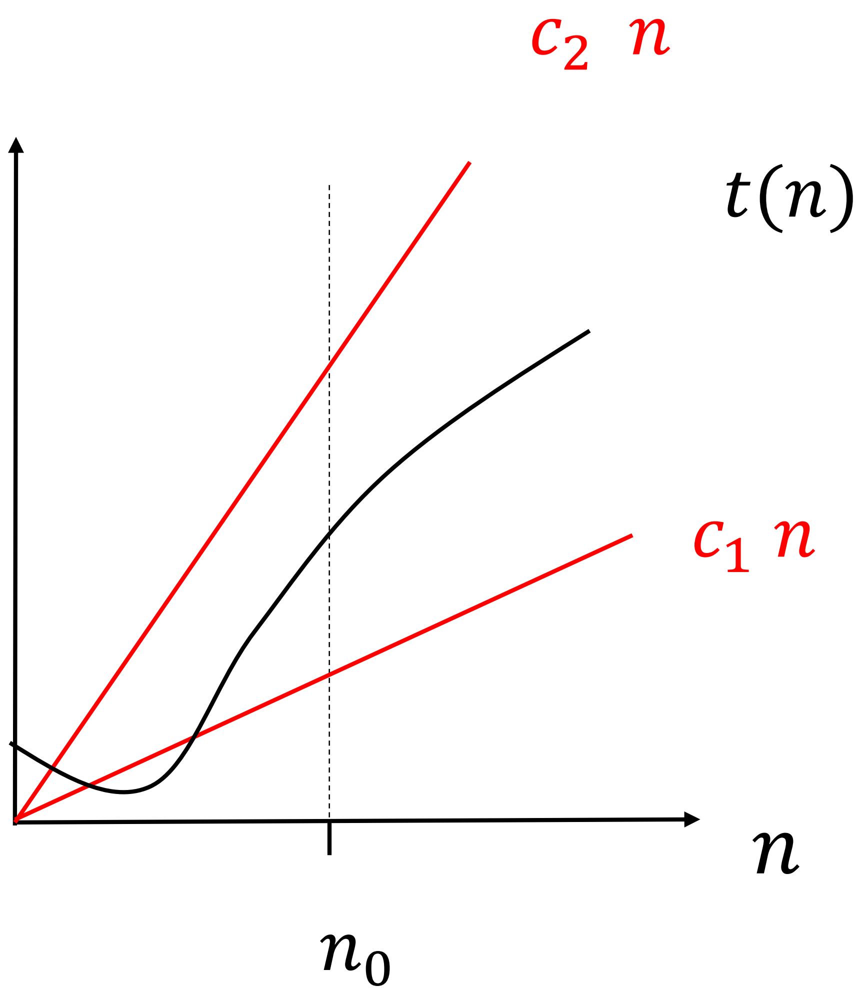
if $t(n)$ is *both* $\mathrm{O}(\, g(n)\,)$ and $\Omega(\, g(n)\,)$.

namely, if there exist three positive constants n_0 , c_1, c_2

such that, for all $n \geq n_0$,

$$c_1 \; g(\, n\,) \leq \; t(\, n\,) \; \leq \; c_2 \; g(\, n\,).$$

Example: $t(n)$ is $\Theta(\ n\ )$



Example

dominant term ↓

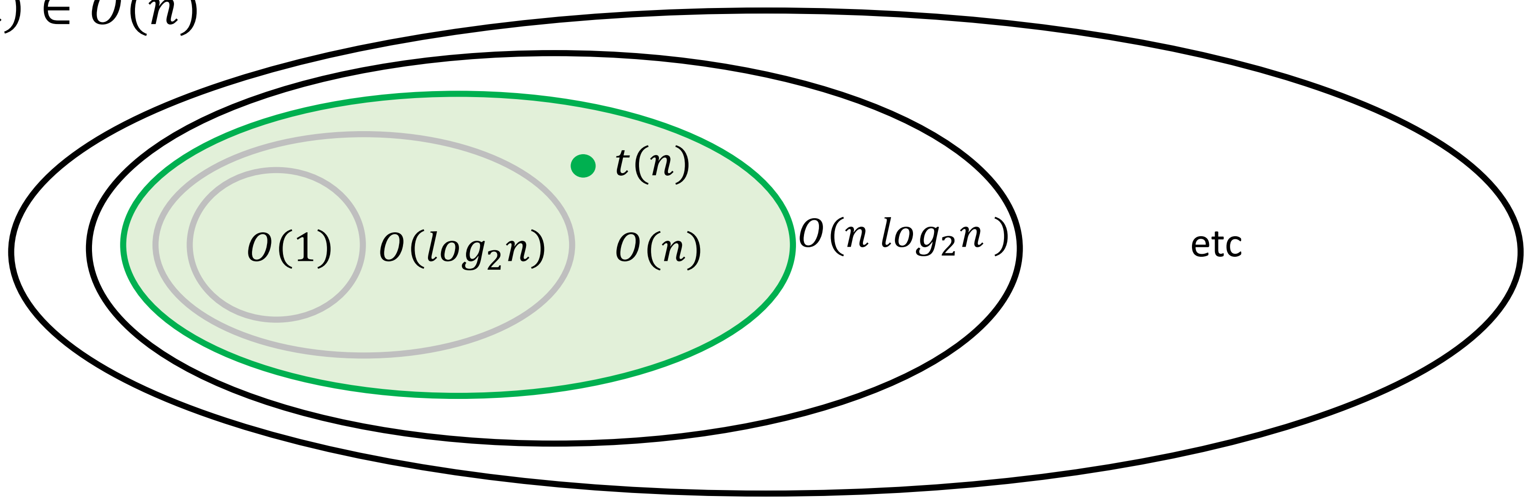
Let $t(n)$ = $4 + 17\, log_2\, n + 3n + 9\, n\, log_2\, n + \boldsymbol{\frac{n(n-1)}{2}}$

Claim: $t(n)$ is $\Theta(\ n^2\ )$.

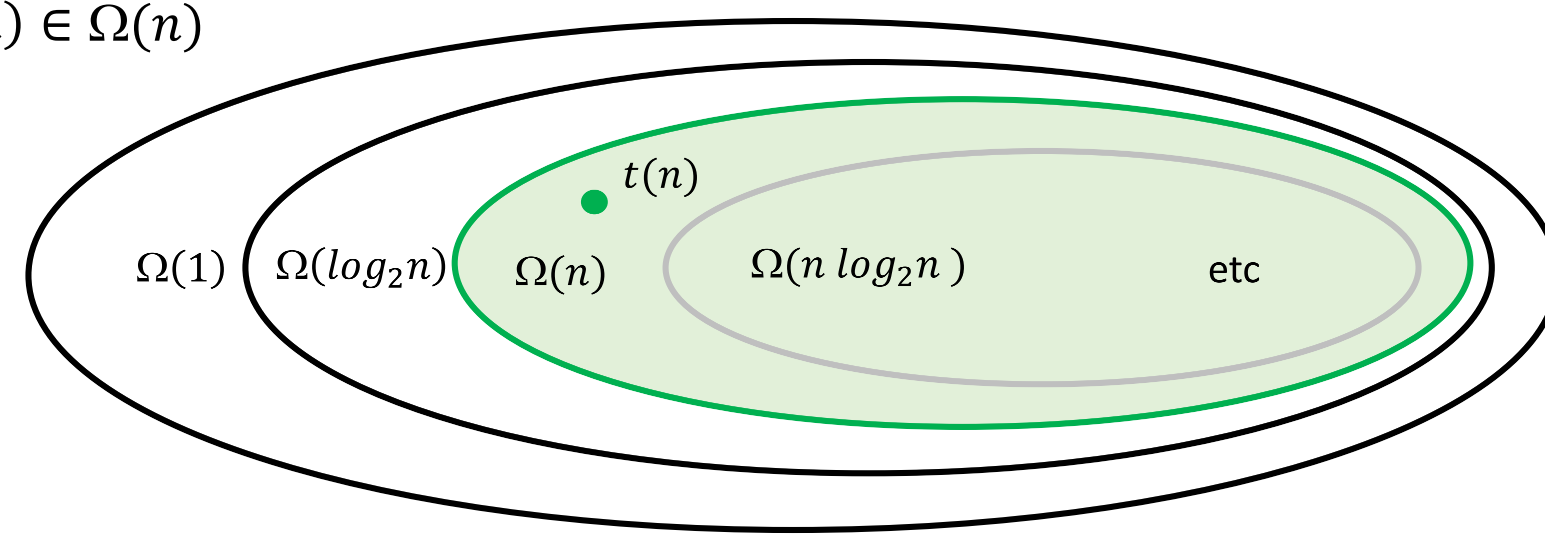
We can prove it by applying the formal definitions of O( ) and Ω( ) .
Details omitted.

Recall last lectures: O, $\boldsymbol{\Omega}$ sets

$t(n) \in O(n)$



$t(n) \in \Omega(n)$

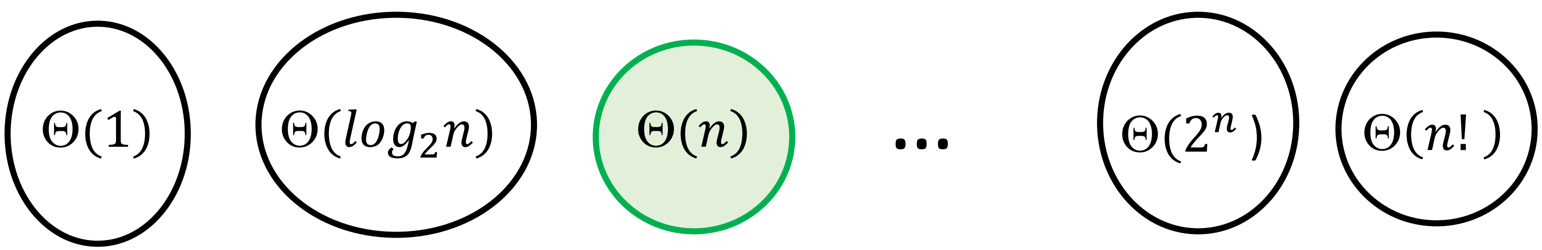


Sets of Θ ( ) functions

If $t(n)$ is $\Theta(\ g(n)\ )$, we often write $t(n) \in \Theta(\ g(n)\ )$,

That is, $t(n)$ is a member of the set of functions that are $\Theta(\ g(n)\ )$.

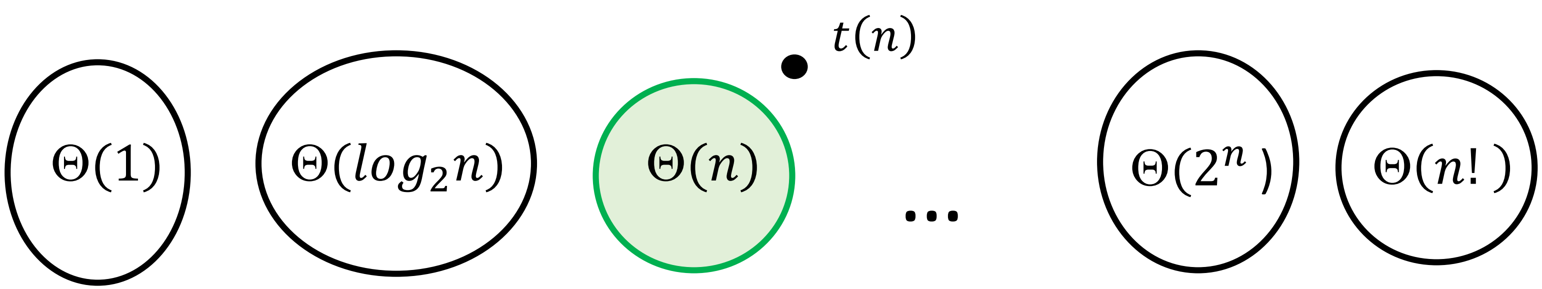
These sets are disjoint.



The funny geometry of the shapes here are just meant to convey that we are taking *the intersection of a* big O set and a big Omega set, *which I have illustrated on the previous slide as ellipsoids*. Do not attach any other significance to these funny shapes!

The figure below suggests that there are functions $t(n)$ that don't belong to any of the $\Theta(\ g(n)\ )$ sets.

What is an example of such a function?



Here is an example of a function that doesn't belong to any of the $\Theta(\ g(n)\ )$ sets :

$$\text{Let } t(n) = \begin{cases} n, & n \text{ is even} \\ 5, & n \text{ is odd.} \end{cases}$$

$t(n)$ is in $O(n\ )$ and $\Omega(1\ )$.

But $t(n)$ is in neither $O(1\ )$ nor $\Omega(\ n\ )$.

Q: The functions $t(n)$ that we care about in this course all belong to some $\Theta(\ )$.

So why are we talking about $O(\ )$ and $\Omega(\ )$?

A: *We sometimes want to discuss upper bounds or lower bounds for an algorithm over all its inputs.*

For examples, when we are discussing a best case we typically have in mind an lower bound $\Omega(\ )$, and when we are discussing a worst case we typically have in mind an upper bound $O(\ )$, respectively.

COMP 250

Lecture 37

big Theta Θ
best and worst cases
limit rules

April 8, 2022

Best and Worst Cases

The time* it takes for an algorithm to run depends on:

- the size n of the input

- the values of the input ← **best versus worst case**

- constant factors

  (#instructions, CPU, programming language)

* As we have seen, "time" could be measured in number of instructions, or number of particular operations, etc.

For some algorithm, suppose the input size is n.

Let $\boldsymbol{t_{best}(n)}$ be the time taken for the best case input.

Let $\boldsymbol{t_{worst}(n)}$ be the time taken for the worse case input.

These are *specific* functions, so they have a *specific* $\Theta(\ )$ behavior.

For $\boldsymbol{t_{best}(n)}$, it is *common* to say $\Omega(\ )$ or $\Theta(\ )$, but not $O(\ )$.

For $\boldsymbol{t_{worst}(n)}$, it is *common* to say $O(\ )$ or $\Theta(\ )$, but not $\Omega(\ )$.

One typically does not talk about an upper bound on the best case, or a lower bound on the worst case, although it would still be correct to do so.

Example of best & worst cases

`Arraylist.remove(i)`

In the best case, `i == size-1` and so the operation takes constant time. So,

$$t_{best}(n) \text{ is } \Omega(1) \text{ or } \Theta(1)$$

In the worst case, `i == 0` and all elements must be shifted. So,

$$t_{worst}(n) \text{ is } O(n) \text{ or } \Theta(n).$$

Recall Binary Search Tree Complexity
(lecture 26)

In the earlier lecture, we used O( ) for best case. But it would make more sense to say $\Omega(\,)$ for best case, if we are emphasizing (tight) lower bound.

	best case	worst case
find(key)	$\Omega(1)$	$O(n)$
findMin()	$\Omega(1)$	$O(n)$
findMax()	$\Omega(1)$	$O(n)$
add(key)	$\Omega(1)$	$O(n)$
remove(key)	$\Omega(1)$	$O(n)$

Recall that best and worst cases are different for each.

Recall Binary Search Tree Complexity
(lecture 26)

If we don't want to emphasize upper and lower bound, and instead we just want to characterize the function, then we can use $\Theta(\,)$.

	best case	worst case
find(key)	$\Theta(1)$	$\Theta(n)$
findMin()	$\Theta(1)$	$\Theta(n)$
findMax()	$\Theta(1)$	$\Theta(n)$
add(key)	$\Theta(1)$	$\Theta(n)$
remove(key)	$\Theta(1)$	$\Theta(n)$

Example: Best and worst case for Lists

	$t_{best}(n)$	$t_{worst}(n)$
add, remove, find an element **(array list or linkedlist)**	$\Theta(1)$	$\Theta(n)$
insertion sort	$\Theta(n)$	$\Theta(n^2)$
selection sort	$\Theta(n^2)$	$\Theta(n^2)$
binary search (sorted array list)	$\Theta(1)$	$\Theta(\log n)$
mergesort	$\Theta(n \log n)$	$\Theta(n \log n)$
quicksort	$\Theta(n \log n)$	$\Theta(n^2)$

best = worst

COMP 250

Lecture 37

big Theta Θ
best and worst cases
limit rules

April 8, 2022

Q: Can we use limits to prove the O, Ω, Θ behavior of a function $t(n)$?

A: Yes, if we apply certain rules.

Limit Rules: Case 1a

Suppose we have $t(n)$ and $g(n)$.

If $$\lim_{n \to \infty} \frac{t(n)}{g(n)} = 0$$

then $t(n)$ is $\mathrm{O}(\, g(n) \,)$.

Why? I will sketch the proof on the next two slides.

Why? Recall definition of Big O

Let $t(n)$ and $g(n)$ be two functions, where $n \geq 0$.

We say $t(n)$ is $\mathrm{O}(\, g(n) \,)$, if there exist two positive constants n_0 and c such that, for all $n \geq n_0$,

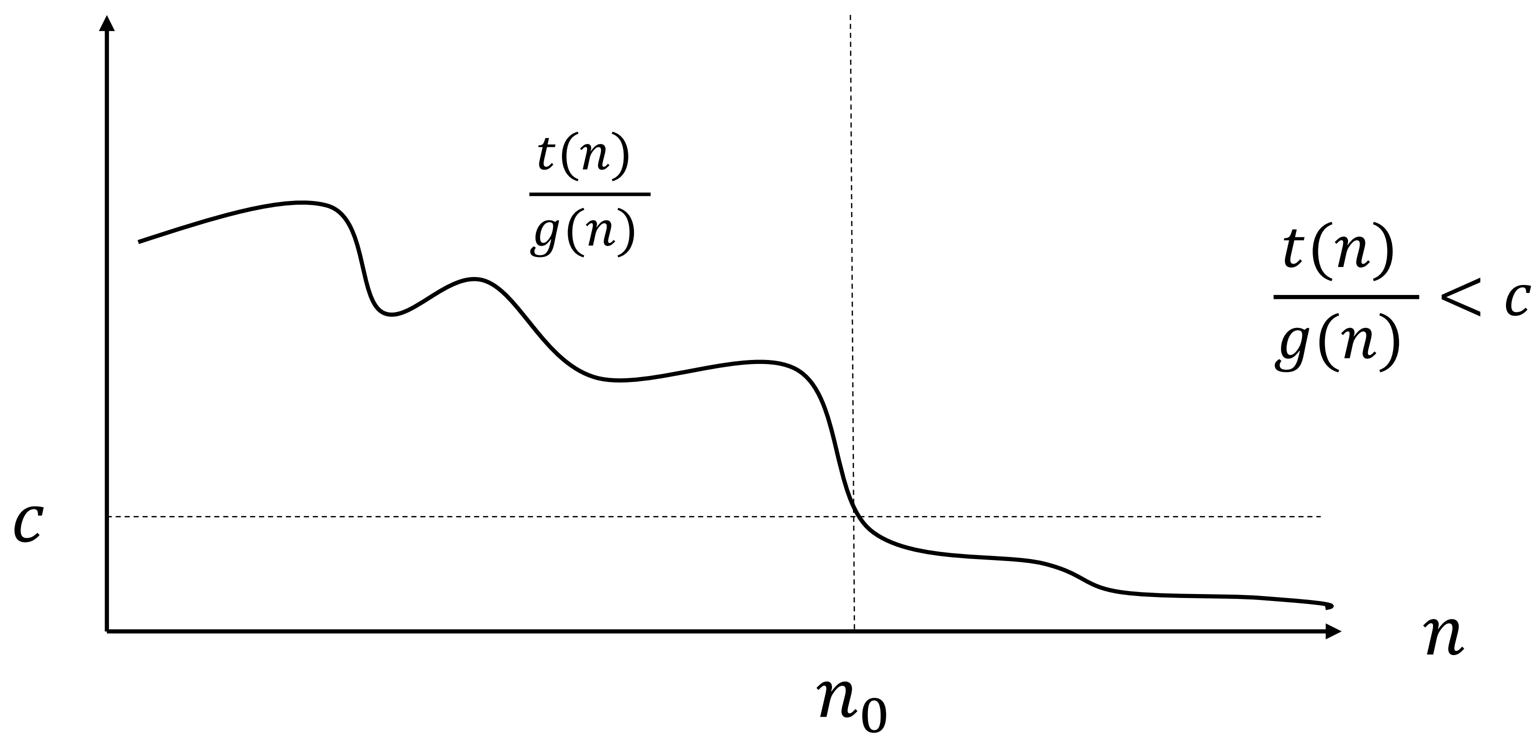
$$t(\, n \,) \leq \; c \;\; g(\, n \,)$$

or equivalently

$$\frac{\boldsymbol{t(n)}}{\boldsymbol{g(n)}} \leq \; \boldsymbol{c} \;\; .$$

Suppose that: $$\lim_{n \to \infty} \frac{t(n)}{g(n)} = 0$$

It follows from the formal definition of a limit (lecture 35) that, for any $c > 0$, $\frac{t(n)}{g(n)}$ will become less than c when n is large enough. This implies that $t(n)$ is $\mathrm{O}(\, g(n)\,)$.



What about the opposite statement (converse)?

If $t(n)$ is $\mathrm{O}(g(n))$ then $\lim_{n \to \infty} \frac{\boldsymbol{t(n)}}{\boldsymbol{g(n)}} = \mathbf{0}$????

No ! For example, take $t(n) = g(n)$.

Then $t(n)$ is $\mathrm{O}(g(n))$, but $\frac{\boldsymbol{t(n)}}{\boldsymbol{g(n)}} = 1$ for all n.

Limit Rules: Case 1b

If $$\lim_{n \to \infty} \frac{t(n)}{g(n)} = 0$$

then $t(n)$ is $O(\, g(n) \,)$

But $t(n)$ is not $\Omega(\, g(n\,))$.

Thus, $t(n)$ is not $\Theta(\, g(n\,))$.

Proof is on the next slide (by contradiction).

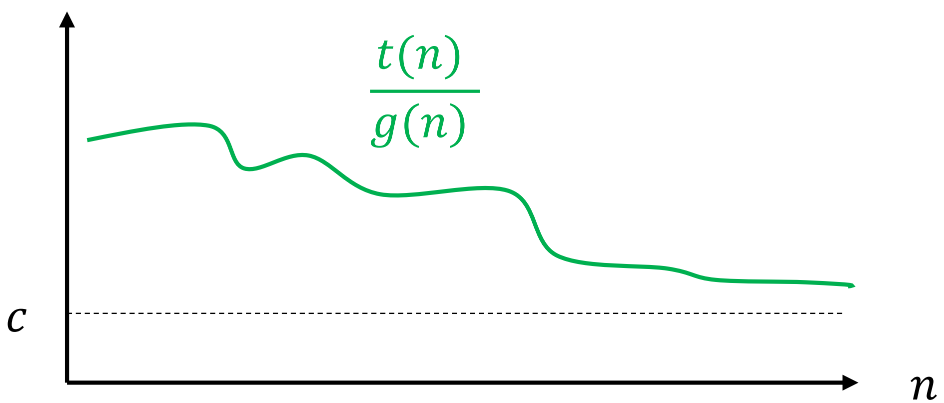
By definition, "$t(n)$ is $\Omega(\ g(n)\ )$" means that:

there exist two constants n_0 and $c > 0$ such that,

for all $n \geq n_0$, $t(n) \geq c\ g(n)$, or equivalently $\dfrac{t(n)}{g(n)} \geq c$.

But this would directly contradict the fact that:

$$\lim_{n \to \infty} \frac{t(n)}{g(n)} = 0$$



Limit Rules: Summary of Case 1

If $$\lim_{n \to \infty} \frac{t(n)}{g(n)} = 0$$

then: $t(n)$ is $O(g(n))$ (1a)

$t(n)$ is not $\Omega(g(n))$ (1b)

Thus, $t(n)$ is not $\Theta(g(n))$.

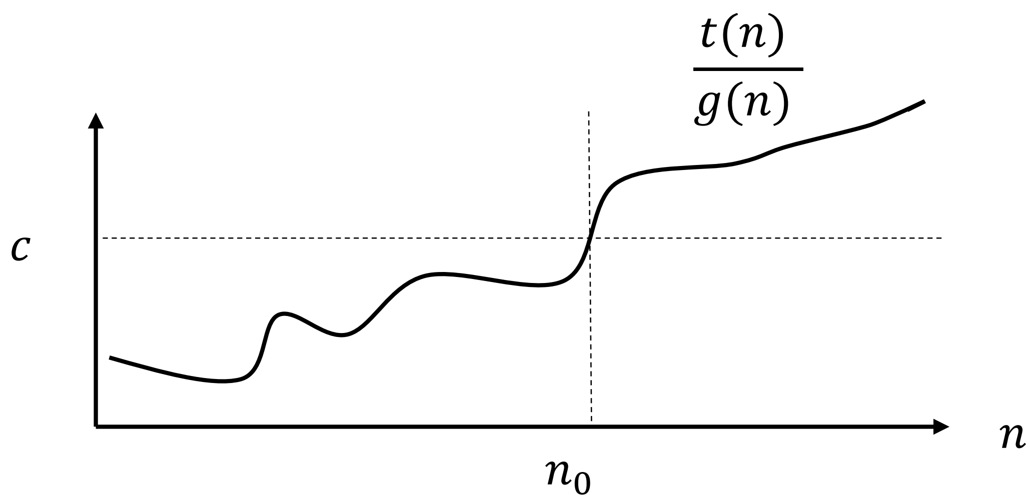
Limit Rules: Case 2

If $$\lim_{\boldsymbol{n} \to \infty} \frac{\boldsymbol{t(n)}}{\boldsymbol{g(n)}} = \infty$$

then:

$t(n)$ is … ?

$t(n)$ is not … ?

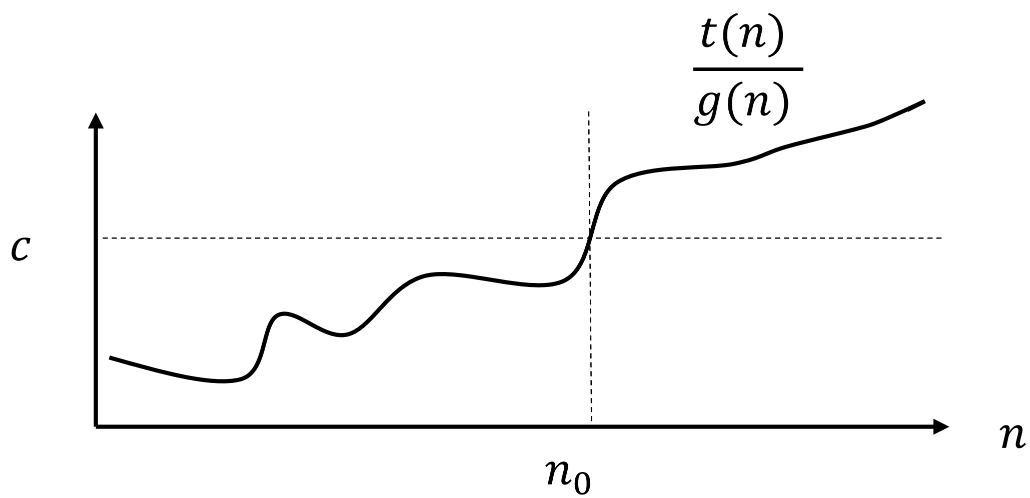


Limit Rules: Case 2

If $\lim_{n \to \infty} \frac{t(n)}{g(n)} = \infty$

then: $t(n)$ is $\Omega(g(n))$.

$t(n)$ is not $O(g(n))$



BTW, some equivalent statements…

$$\lim_{n \to \infty} \frac{t(n)}{g(n)} = \infty \quad \Leftrightarrow \quad \lim_{n \to \infty} \frac{g(n)}{t(n)} = 0$$

$t(n)$ is $\Omega(g(n))$ $\Leftrightarrow$ $g(n)$ is $O(t(n))$

$t(n)$ is not $O(g(n))$ $\Leftrightarrow$ $g(n)$ is not $\Omega(t(n))$

Limit Rules: Case 3

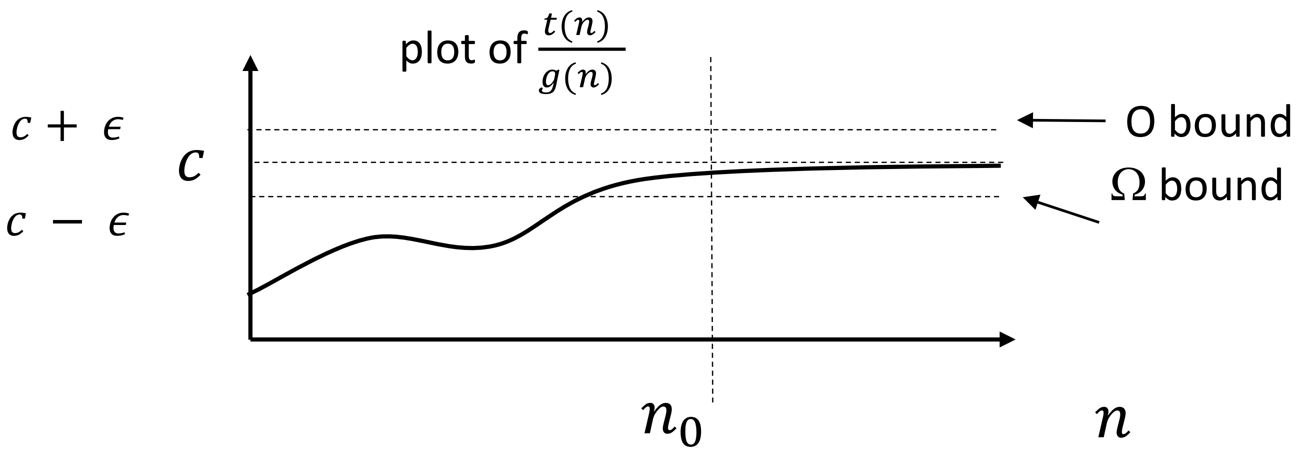
If

$$\lim_{n \to \infty} \frac{t(n)}{g(n)} = c \,, \quad 0 < c < \infty$$

then

$$t(n) \text{ is } \Theta(\, g(n\,)).$$

Proof (sketch only):



Limit Rules

All three rules just discussed say that *if* a limit exists:

$$\lim_{n \to \infty} \frac{t(n)}{g(n)} \text{ \textit{is 0, ∞, or} } c > 0$$

then we can say something about the O, Ω, Θ relationship between $t(n)$ and $g(n)$.

However, *if* the limit does *not* exist, then the limit rules do *not* tell us anything.

Final Exam

- Closed book.   No crib sheet.   No calculators.
- 45 Questions.   Multiple choice: 4 choices per question.
- Do not leave any questions blank!
- If you get 20/45 or worse, then the highest grade you can get in course is D.   **See grading policy on the Course Outline.**

Final Exam – how to prepare?

- review lectures that you didn't quite understand
- do the exercise PDFs
- do practice quizzes
- do *not* review the assignments

Thinking about Graduate School ?

I will add to mycourses a lecture from Fall where I talk about CS graduate school (MSc, PhD).

- why or why not get a graduate degree (in CS) ?
- my experience(s)
- research life
- MSc vs. PhD, preparations
- equity, diversity, inclusion

Please fill out Mercury Course Evaluations.

MERCURY
COURSE EVALUATIONS

I plan to have office hours on April 29 & 20.

I will be on the discussion board in the meantime.

Good luck with studying !