An Algorithmist's Toolkit | Brown CSCI 2952T – Spring 2026

Lecture 1: Asymptotics

1/21/26 | Lecturer: D Ellis Hershkowitz | Scribe: Maryam Abuissa

1 Asymptotic Hierarchy

So we've decided we like *simplification*. Let's see a technical embodiment of that ideal.

Often in theory, the time or space complexity of an algorithm is a function of the input size n. For example, you may have an algorithm that runs in time

$$2n\log(n) + \frac{e^{2n}}{n+1}. \tag{1}$$

That's cool and all, but it's also complicated and confusing.

In CS theory, we like to squint. But of course, we still have to be rigorous. We rigorously squint using *asymptotic hierarchies*. This keeps functions simple.

What do we mean by simple? Well the simplest function of n is $f(n) = n$.

Modifications that preserve simplicity of a function:

- multiplication of simple functions
- powers of a function
- exponentiation (and logarithms) of a function

For example, something like

$$n\log(n)n^{\log(n)^2} \tag{2}$$

is actually pretty simple, even though it looks gross. The intuition is that eq. (2) is as simple as it can be in order to preserve its general behavior.

Note that our simplicity-preserving modifications didn't include constants and addition. How do we get rid of those things?

1.1 Big O + friends

"Big O notation" primarily gets rid of constants, although it also sometimes gets rid of additive terms.

Definition 1 (Big O). *$f(n) = O(g(n))$ if $\exists c, n_0$ such that $\forall n \geq n_0$,*

$$|f(n)| \leq c * g(n) \tag{3}$$

Note: In CS, sometimes we omit the absolute value because it usually doesn't matter for us.

We will use the concept of big O a lot, so we need to be able to modify it in a few ways.

Definition 2 (Function of big O). *$f(n) = g(O(h(n)))$ if $\exists h'(n)$ such that $h'(n) = O(h(n))$ and $f(n) = g(h'(n))$.*

Sometimes we want a parameter which tends to 0 for our asymptotic analysis. In this case we will use ϵ instead of n.

Definition 3 (Big O for ϵ). *$f(\epsilon) = O(g(\epsilon))$ if $\exists c, \epsilon_0$ such that $\forall 0 \leq \epsilon \leq \epsilon_0$ have*

$$|f(\epsilon)| \leq c * g(\epsilon) \tag{4}$$

These can get confusing. The least ambiguous notation uses a subscript of big O to tell us what variable is tending to what, e.g.

$$O_{\epsilon \to 0}\left(\frac{1}{x^2}\right) \tag{5}$$

1.1.1 Inequality analogy

Here is where the friends come in. Although we can't usually compare functions directly with inequalities, the following analogy shows how each asymptotic relation intuitively corresponds to a particular inequality relation.

Asymptotic relation	**Inequality relation**
$f = O(g)$	$f \leq g$
$f = \Omega(g)$	$f \geq g$
$f = \Theta(g)$	$f = g$
$f = o(g)$	$f < g$
$f = \omega(g)$	$f > g$

We will now formally define each of the asymptotic relations above.

Definition 4 (Big Ω). *$f(n) = \Omega(g(n))$ if $\exists c > 0, n_0$ such that $\forall n \geq n_0$,*

$$|f(n)| \geq c * g(n) \tag{6}$$

Definition 5 (Big Θ). *$f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.*

Definition 6 (Little o). *$f(n) = o(g(n))$ if*

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \tag{7}$$

Definition 7 (Little ω). *$f(n) = \omega(g(n))$ if*

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \tag{8}$$

1.1.2 Other asymptotic notation

Sometimes we want to define other asymptotic relationships.

Note: we will sometimes simplify functions $f(n)$ to f.

Definition 8 ("Poly": $\text{poly}(\cdot)$)**.** *$f(n) = \text{poly}(g(n))$ if $\exists c > 0$ such that*

$$f = O(g(n)^c) \tag{9}$$

The next few definitions effectively *omit logarithmic factors* which is useful when they are much smaller than the other terms and therefore not as important.

Definition 9 ("Big O tilde": $\tilde{O}(\cdot)$)**.** *$f(n) = \tilde{O}(g(n))$ if $f = O(h * g)$ for some*

$$h = \text{poly}(\log(n)) \tag{10}$$

Definition 10 ("Big Ω tilde": $\tilde{\Omega}(\cdot)$)**.** *$f(n) = \tilde{\Omega}(g(n))$ if $f = \Omega(h * g)$ for some*

$$h = \frac{1}{\text{poly}(\log(g))} \tag{11}$$

2 Eliminating Troublemakers

So we don't like constants or addition. Asymptotic notation takes away constants and some additive terms. What about the tricky ones?

2.1 Dealing with $1 + x$

"The most important theorem in theoretical CS." -Ellis

Theorem 1. *For "small" x,*

$$1 + x \approx e^x \tag{12}$$

We will break this into two lemmas.

Lemma 1.

$$1 + x < e^x \tag{13}$$

Proof. We will prove by cases.

Case 1. $x \leq -1$.

Then $1 + x \leq 0$ but $e^x \geq 0$.

Case 2. $x \geq 0$

Then

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \geq 1 + x \tag{14}$$

Case 3. $x \in (-1, 0)$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \geq 1 + x \tag{15}$$

This is because for $x \in (0, -1)$, $\frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \geq 0$ because $\frac{x^2}{2!} + \frac{x^3}{3!} \geq \frac{x^2 + x^3}{3!} \geq 0$, and so on and so forth for the bigger exponents. □

Lemma 2.

$$e^x \leq 1 + x + \Theta_{x \to 0}(x^2) \tag{16}$$

Proof. Using the Taylor series expansion of e^x, we have

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = 1 + x + \Theta_{x \to 0}(x^2) \tag{17}$$

□

Note that theorem 1 makes a claim about small x, which applies as $x \to 0$. What if x is getting big instead of small??

Well, then $\frac{1}{x}$ is getting small!

This idea allows us to do cool tricks like:

$$\ln(n+1) = \ln(n(1 + \frac{1}{n})) \tag{18}$$

$$= \ln(n) + \ln(1 + \frac{1}{n}) \tag{19}$$

$$\approx \ln(n) + \frac{1}{n} \tag{20}$$

where we used theorem 1 for eq. (20).

2.2 Dealing with factorials

Factorials are quite common, and yet they suck. They are not simple, nor are they easy to squint at. Luckily we have a theorem for this.

Theorem 2.

$$n! \approx \left(\frac{n}{e}\right)^n \tag{21}$$

Again, we will break this up into 2 lemmas.

Lemma 3.

$$n! \geq \frac{n^n}{e^n} \tag{22}$$

Proof. Again we'll use the Taylor expansion of e^x.

$$e^n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{n^n}{n!} \geq \frac{n^n}{n!} \tag{23}$$

□

It's quite easy to show that $n! \leq n^n$, since $n! = n(n-1)(n-2)\cdots \leq n^n$. This is sort of close to what we want, but we can do one better.

Lemma 4.

$$n! \leq O(n)\left(\frac{n}{e}\right)^n \tag{24}$$

Not that this is basically what we want to show since n is waaaaay smaller than a factorial, and therefore doesn't matter in the context of algorithms.

Proof. Mysteriously, sometimes we *like* sums because they allow us to do calculus. We will use this fact for the proof. We start by taking logs of both sides.

$$\begin{aligned}
\ln(n!) &= \sum_{i=1}^{n} \ln(i) && (25)\\
&\leq \int_{i=1}^{n+1} (\ln(i))\delta i && (26)\\
&= i\ln(i) - i\Big|_{1}^{n+1} && (27)\\
&= (n+1)\ln(n+1) - n && (28)\\
&\leq (n+1)\left(\ln(n) + \frac{1}{n}\right) - n && (29)\\
&= n\ln(n) + \ln(n) - n + O(1) && (30)
\end{aligned}$$

Now we can undo the logs to get back to

$$\begin{aligned}
n! &\leq \exp(n\ln(n) + \ln(n) - n + O(1)) && (31)\\
&= \frac{n^n * n}{e^n} O(1) = \left(\frac{n}{e}\right)^n O(n) && (32)
\end{aligned}$$

□

2.3 Dealing with chooses

When you're doing algorithms, n choose k will also pop up often. This is even worse to squint at, with so many factorials in its formula.

Theorem 3.

$$\binom{n}{k} \approx \left(\frac{ne}{k}\right)^k \tag{33}$$

Once more, we'll split it into two lemmas.

Lemma 5.

$$\binom{n}{k} \leq \left(\frac{ne}{k}\right)^k \tag{34}$$

Proof. Here we'll just use algebra and lemma 3.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \leq \frac{n^k}{k!} \leq \frac{n^k}{\left(\frac{k}{e}\right)^k} = \left(\frac{ne}{k}\right)^k \tag{35}$$

□

Lemma 6.

$$\binom{n}{k} \geq \left(\frac{n}{k}\right)^k \tag{36}$$

Proof. We'll start with expanding out the factorials in the definition.

$$\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots(1)} \geq \left(\frac{n}{k}\right)^k \tag{37}$$

The final inequality uses the fact that $\forall i$,

$$\frac{n-i}{k-i} \geq \frac{n}{k}$$

To see this fact, note that $n \geq k$ implies that $(n-i)k \geq n(k-i)$ (because algebra - exercise). □