

University of Waterloo
CS240, Fall 2026
Module 1 **Math Review and Intuition**

Xuezhi (Jerry) Wang, CS240 ISA

1 Do I really need the exact value?

It's 4:49pm¹ on a Tuesday, and the assignment is due in 10 minutes. You are solving the last problem on the assignment, but the time complexity just doesn't look right. You stare at the ugly summation:

$$n \cdot \sum_{i=1}^n \frac{i^{420} \log i}{69^i + i^{67}}$$

"How do I even begin to resolve this?" you ask yourself.

The better question to ask here is, "Do I need to solve this?" If the question is asking for **asymptotic notation**, then whether the constant is big or small does not matter.

Look at the sum again. Is its limit as $n \rightarrow \infty$ a constant? A useful tool might be the **ratio test** from Calculus. If the limit exists, you can set the limit to a constant L , and bound the sum above by

$$n \cdot \sum_{i=1}^n \frac{i^{420} \log i}{69^i + i^{67}} \leq nL \in O(n)$$

There are a variety of other tests, such as **root test** / **integral test** which may help you resolve the summation. If you need more details, refer to prior Calculus (MATH137/147) notes.

2 What if I need it?

There are a few common approaches. One of them is **telescoping**, where you engineer consecutive terms to cancel. Suppose you are trying to find the sum:

$$S_n = \sum_{i=1}^n \frac{i}{2^i}$$

¹For all you procrastinators out there, 7:49pm. Please don't use the grace period for doing assignments.

$$\begin{aligned}
S_n &= 2S_n - S_n = \sum_{i=1}^n \frac{i}{2^{i-1}} - \sum_{i=1}^n \frac{i}{2^i} \\
&= 1 + \left(\sum_{i=1}^{n-1} \frac{i+1}{2^i} - \frac{i}{2^i} \right) + \frac{i}{2^n} \\
&= 1 + \left(\sum_{i=1}^{n-1} \frac{1}{2^i} \right) + \frac{i}{2^n} \\
&= 1 + \left(1 - \frac{1}{2^{n-1}} \right) - \frac{i}{2^n} \\
&= 2 - \frac{i+2}{2^n}
\end{aligned}$$

Another common approach is to think about where you've seen the sum before. There are some famous sums, such as the Basel problem: $\zeta(2) = \sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$.

3 What is my induction hypothesis?

Induction is a powerful tool and can be used to resolve recursions, but coming up with an induction hypothesis sometimes seems **magical**. Suppose we have the problem

$$T(n) = 8T(n/2) + n^2$$

And we want to show that $T(n) \in O(n^3)$.

A natural thing to try is $T(n) \leq c_1 n^3$. Suppose this holds for all $n < k$.

$$T(k) = 8T\left(\frac{k}{2}\right) + k^2 \leq c_1 k^3 + k^2$$

Wait, what? There's a pesky k^2 term there.

But we can modify our induction hypothesis to be a bit stronger. Suppose $T(n) \leq c_1 n^3 + c_2 n^2$ for all $n < k$. We don't know what c_2 is yet, but that's ok. We can **find** the constant c_2 that works for our problem.

$$T(k) = 8T\left(\frac{k}{2}\right) + 2c_2 k^2 + k^2 \leq c_1 k^3 + (2c_2 + 1)k^2$$

Now, we realize that as long as $c_2 \geq 2c_2 + 1$, the induction hypothesis holds! Solving for c_2 , we get $c_2 \leq -1$. So we can use the induction hypothesis of $T(n) = c_1 n^3 - n^2$.

In the writeup, it is recommended that you include the reasoning. However, this is why sometimes you see "magical" hypotheses.

4 Dealing with Human-made Slop

The course is supposed to be rigorous! Why do I see “sloppy” stuff everywhere? Missing floors and ceilings like the M4 building², $\Theta(n) \ni f(n) \approx cn$, how do these make sense?

4.1 Sloppy Recurrence

This is honestly a bit of a misnomer. “Sloppy” isn’t describing the recurrence. It’s describing **a restriction on n** .

Take the recurrence $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + 1$. If we assume $n = 2^k$, then every single time we apply the recurrence, $\lfloor \frac{n}{2} \rfloor = \frac{n}{2}$.

Think about what conditions you need to generalize $f(n) \in O(g(n))$ from $\forall n = 2^k$ to $\forall n \in \mathbb{N}$. What about $n = 2^{2^k}$? The following are good to consider:

- **Monotonicity**. If the function isn’t non-decreasing, the bounds on $f(2^k)$ can’t be applied to the remaining integers.
- **Growth**. Even if the function is strictly increasing, if the values for $f(n)$ and $f(2n)$ differ a lot, maybe more than a constant factor, is it still $O(g(n))$ everywhere?

However, in a lot of cases, the generalization step from $n = 2^k$ to $n \in \mathbb{N}$ holds. This is why we sometimes use sloppy recurrences.

4.2 Sloppy Notation

Sometimes, in the textbook, you may see notation like $f(n) \in \Theta(g(n))$, so we let $f(n) \approx cg(n)$. What is this $\approx$ doing here?

This c is **hiding two constants c_1, c_2** . For sufficiently large $n > N$,

$$f(n) \approx cg(n) \iff c_1g(n) \leq f(n) \leq c_2g(n)$$

It is mostly meant as a shorthand and for convenience. In general, avoid working with asymptotic notation when doing algebraic manipulations. Better to replace $f(n) \in \Theta(g(n))$ with $f(n) = c \cdot g(n)$, while being aware that there are two constants c_1, c_2 which both needs to be preserved, and n needs to be sufficiently large.

²When it started construction