Let \( a \geq 1 \) and \( b > 1 \) be integers, and let \( f(n) \) be a non-negative function. The master method provides solutions for many recursions of the form

\[
T(n) = aT\left(\frac{n}{b}\right) + f(n)
\]
(with any constant-time base case).

**Case 1**: if \( f(n) = O(n^{\log_b a - \epsilon}) \) for some constant \( \epsilon > 0 \), then

\[
T(n) = \Theta(n^{\log_b a}).
\]

**Case 2**: if \( f(n) = \Theta(n^{\log_b a \log^k n}) \), then

\[
T(n) = \Theta(n^{\log_b a \log^{k+1} n}).
\]

**Case 3**: if \( f(n) = \Omega(n^{\log_b a + \epsilon}) \) for some constant \( \epsilon > 0 \), and “\( f \) is not strange,” then

\[
T(n) = \Theta(f(n)).
\]

By “\( f \) is not strange,” we mean that \( f \) satisfies the following *regularity condition* with respect to the \( a \) and \( b \) appearing in the recurrence: for some constant \( c < 1 \) and all sufficiently large \( n \),

\[
a f(n/b) \leq cf(n).
\]

This condition is rarely violated in practice. In particular, it is satisfied for any \( f(n) \) that is a polynomial or looks like \( cn^j \log^k n \) for any \( c, j, k \geq 0 \).
In class, I claimed that polynomials are not strange. Here’s a more precise statement with the proof:

**Lemma:** Suppose that we have a recurrence of the form

\[ T(n) = aT\left( \frac{n}{b} \right) + n^j \]

to which the third case of the master method above might apply; that is, \( j = \log_b a + \epsilon \) for some constant \( \epsilon > 0 \). Then

\[ a \left( \frac{n}{b} \right)^j = \frac{1}{b^\epsilon} n^j. \]

Since \( b > 1 \), this inequality proves that \( n^j \) does not violate the regularity condition.

**Proof:**

\[
\begin{align*}
    a \left( \frac{n}{b} \right)^j &= \frac{a n^j}{b^j} \\
    &= \frac{a n^j}{b^{\log_b a + \epsilon}} \\
    &= \frac{a n^j}{b^{\log_b a} \cdot b^{\epsilon}} \\
    &= \frac{a n^j}{a b^\epsilon} \\
    &= \frac{n^j}{b^\epsilon} \quad \text{QED.}
\end{align*}
\]

Technically, this proof only covers monomials, but it’s not too hard to show that the sum or difference of two monomials isn’t strange either (provided that the latter is non-negative, of course), and we can then expand non-strangeness inductively to arbitrary polynomials.

It’s also not hard to show that multiplying \( n^j \) by \( \log^k n \) for any \( k \) doesn’t cause it to become strange:

**Lemma:** Suppose that we have a recurrence of the form

\[ T(n) = aT\left( \frac{n}{b} \right) + n^j \log^k n \]

to which the third case of the master method above might apply; that is, \( k = \log_b a + \epsilon \) for some constant \( \epsilon > 0 \). Then

\[ a \left( \frac{n}{b} \right)^j \log^k \left( \frac{n}{b} \right) \leq \frac{1}{b^\epsilon} n^j \log^k n. \]

**Proof:**

\[
\begin{align*}
    a \left( \frac{n}{b} \right)^j \log^k \left( \frac{n}{b} \right) &= \frac{a n^j}{b^j} \log^k \left( \frac{n}{b} \right) \\
    &= \frac{n^j}{b^\epsilon} (\log n - \log b)^k \\
    &\leq \frac{1}{b^\epsilon} n^j \log^k n \quad \text{QED.}
\end{align*}
\]

Hence, it’s rather difficult to find recurrences with strange functions in the practical analysis of algorithms.

Just to show you that there are some strange functions out there, consider the recurrence

\[ T(n) = T\left( \frac{n}{2} \right) + 4^{\lceil \log_4 n \rceil}. \]

For this recurrence, \( a = 1 \) and \( b = 2 \), and it’s not hard to see that \( 4^{\lceil \log_4 n \rceil} \geq n \), which is polynomially greater than \( n^{\log_2 4} = 4 \). However, \( 4^{\lceil \log_4 (n/2) \rceil} = 4^{\lceil \log_4 n \rceil} \) whenever \( n \) is a power of 4, so the regularity condition is violated.