LECTURE 9
Solving recurrences
• Master theorem
Review questions

• Guess the solution to the recurrence:
  \[ T(n) = 2T(n/3) + n^{3/2}. \]

  (Answer: \( \Theta(n^{3/2}). \))

• Draw the recursion tree for this recurrence.
  a. What is its height?

  (Answer: \( h = \log_3 n. \))

  b. What is the number of leaves in the tree?

  (Answer: \( n^{(1/\log 3)}. \))
The master method

The master method applies to recurrences of the form

\[ T(n) = a \cdot T(n/b) + f(n), \]

where \( a \geq 1 \), \( b > 1 \), and \( f \) is asymptotically positive, that is \( f(n) > 0 \) for all \( n > n_0 \).
Three common cases

Compare $f(n)$ with $n^{\log_b a}$:

1. $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$.
   - $f(n)$ grows polynomially slower than $n^{\log_b a}$ (by an $n^\varepsilon$ factor).

   **Solution:** $T(n) = \Theta(n^{\log_b a})$. 
Three common cases

Compare $f(n)$ with $n^{\log_b a}$:

1. $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$.
   - $f(n)$ grows polynomially slower than $n^{\log_b a}$ (by an $n^\epsilon$ factor).
   
   **Solution:** $T(n) = \Theta(n^{\log_b a})$.

2. $f(n) = \Theta(n^{\log_b a \lg^k n})$ for some constant $k \geq 0$.
   - $f(n)$ and $n^{\log_b a}$ grow at similar rates.
   
   **Solution:** $T(n) = \Theta(n^{\log_b a \lg^{k+1} n})$. 
Three common cases (cont.)

Compare \( f(n) \) with \( n^{\log ba} \):

3. \( f(n) = \Omega(n^{\log ba} + \varepsilon) \) for some constant \( \varepsilon > 0 \).
   - \( f(n) \) grows polynomially faster than \( n^{\log ba} \) (by an \( n^\varepsilon \) factor),
   
and \( f(n) \) satisfies the *regularity condition* that \( af(n/b) \leq cf(n) \) for some constant \( c < 1 \).

*Solution*: \( T(n) = \Theta(f(n)) \).
Idea of master theorem

Recursion tree:

\[ f(n) \]
\[ a \]
\[ \frac{f(n)}{b} \quad \frac{f(n)}{b} \quad \cdots \quad \frac{f(n)}{b} \]
\[ a \]
\[ \frac{f(n/b^2)}{b^2} \quad \frac{f(n/b^2)}{b^2} \quad \cdots \quad \frac{f(n/b^2)}{b^2} \]
\[ / \]
\[ / \]
\[ / \]
\[ T(1) \]
Idea of master theorem

Recursion tree:

\[ f(n) \quad a \quad f(n) \]
\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b) \quad a f(n/b) \]
\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \quad a^2 f(n/b^2) \]
\[ \vdots \]
\[ T(1) \]
Idea of master theorem

Recursion tree:

\[ f(n) \quad \cdots \quad a f(n/b) \]
\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b/b) \]
\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \quad a^2 f(n/b^2) \]

\[ h = \log_b n \]

\[ T(1) \]
Idea of master theorem

Recursion tree:

\[ h = \log_b n \]

\[ f(n) \]

\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b) \quad a f(n/b) \]

\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \quad a^2 f(n/b^2) \]

\[ \text{#leaves} = a^h \]

\[ = a^{\log_b n} \]

\[ = n^{\log_b a} \]

\[ T(1) \]

\[ = n^{\log_b a} T(1) \]
Idea of master theorem

Recursion tree:

\[ f(n) \overset{a}{\longrightarrow} f(n) \]
\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b) \overset{a}{\longrightarrow} af(n/b) \]
\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \overset{a}{\longrightarrow} a^2f(n/b^2) \]
\[ \vdots \]
\[ T(1) \]

\[ h = \log_b n \]

CASE 1: The weight increases geometrically from the root to the leaves. The leaves hold a constant fraction of the total weight.

\[ n^{\log_b a} T(1) \]

\[ \Theta(n^{\log_b a}) \]
Idea of master theorem

Recursion tree:

\[ f(n) \quad a \quad f(n) \]
\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b) \quad a f(n/b) \]
\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \quad a^2 f(n/b^2) \]
\[ \vdots \]
\[ T(1) \quad n^{\log_b a} T(1) \quad \Theta(n^{\log_b a} \log n) \]

CASE 2: \((k = 0)\) The weight is approximately the same on each of the \(\log_b n\) levels.
Idea of master theorem

**Recursion tree:**

\[ h = \log_b n \]

\[ f(n) \]

\[ f(n/b) \quad f(n/b) \quad \cdots \quad f(n/b) \]

\[ a \quad a \quad a \quad \cdots \quad a \]

\[ f(n/b^2) \quad f(n/b^2) \quad \cdots \quad f(n/b^2) \]

\[ a^2 f(n/b^2) \]

**CASE 3:** The weight decreases geometrically from the root to the leaves. The root holds a constant fraction of the total weight.

\[ n^{\log_b a} T(1) \]

\[ \Theta(f(n)) \]
Examples

**Ex.** \( T(n) = 4T(n/2) + n \)

\[ a = 4, \ b = 2 \implies n^{\log_b a} = n^2; \ f(n) = n. \]

**Case 1:** \( f(n) = O(n^{2-\varepsilon}) \) for \( \varepsilon = 1. \)

\[ \therefore T(n) = \Theta(n^2). \]
Examples

Ex. \[ T(n) = 4T(n/2) + n \]
\[ a = 4, \ b = 2 \Rightarrow n^{\log b a} = n^2; \ f(n) = n. \]

**CASE 1:** \[ f(n) = O(n^2 - \varepsilon) \] for \( \varepsilon = 1. \)
\[ \therefore \ T(n) = \Theta(n^2). \]

Ex. \[ T(n) = 4T(n/2) + n^2 \]
\[ a = 4, \ b = 2 \Rightarrow n^{\log b a} = n^2; \ f(n) = n^2. \]

**CASE 2:** \[ f(n) = \Theta(n^2 \lg^0 n), \text{ that is, } k = 0. \]
\[ \therefore \ T(n) = \Theta(n^2 \lg n). \]
Examples

Ex. \( T(n) = 4T(n/2) + n^3 \)
\[ a = 4, \quad b = 2 \Rightarrow n^{\log_b a} = n^2; \quad f(n) = n^3. \]

Case 3: \( f(n) = \Omega(n^2 + \epsilon) \) for \( \epsilon = 1 \)
and \( 4(n/2)^3 \leq cn^3 \) (reg. cond.) for \( c = 1/2. \)
\[ \therefore T(n) = \Theta(n^3). \]
Examples

Ex.  $T(n) = 4T(n/2) + n^3$

$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^3$.

**CASE 3:** $f(n) = \Omega(n^2 + \varepsilon)$ for $\varepsilon = 1$

*and* $4(n/2)^3 \leq cn^3$ (reg. cond.) for $c = 1/2$.

$\therefore T(n) = \Theta(n^3)$.

Ex.  $T(n) = 4T(n/2) + n^2/\lg n$

$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^2/\lg n$.

Master method does not apply. In particular, for every constant $\varepsilon > 0$, we have $n^\varepsilon = \omega(\lg n)$. 