1 Recursion trees

We can solve many recurrences by drawing a recursion tree.

- **Nodes**: Label with the contribution to the total for that ‘recursive call’.
  
  … not counting what happens inside children.

- **Children**: One for each appearance of a recurrent term.

After drawing such a tree, we can solve the recurrence:

1. Compute (or bound) the **depth of the leaves**.
2. Compute (or bound) the **sum for each level**.
3. Compute (or bound) the **sum across all levels**.

2 Example: Mergesort recurrence

\[ T(n) = \begin{cases} 
\Theta(1) & \text{if } n \leq 1 \\
2T\left(\frac{n}{2}\right) + \Theta(n) & \text{otherwise}
\end{cases} \]

\[ cn \]

\[ \frac{cn}{2} \]

\[ \frac{cn}{4} \]

\[ \frac{cn}{4} \]

\[ \frac{cn}{4} \]

\[ \frac{cn}{4} \]

\[ \vdots \]

\[ \Theta(1) \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \Theta(1) \]

3 Example: Mergesort recurrence

- **Depth of the leaves**: \(\lg n\)

- **Sum for each level**: \(cn\)

- **Sum across all levels**: \(cn \lg n = \Theta(n \log n)\).
4 Example: Another divide-and-conquer recurrence

\[ T(n) = \begin{cases} 
\Theta(1) & \text{if } n \leq 1 \\
3T\left(\frac{n}{4}\right) + \Theta(n^2) & \text{otherwise} 
\end{cases} \]

\[ T(n) = \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i cn^2 \]
\[ \leq \frac{1}{1 - (3/16)}cn^2 
\]
\[ = \frac{16}{13}cn^2 
\]
\[ = O(n^2) \]

Note also: \( T(n) = \Omega(n^2) \). (Why?)

At depth \( i \), the ‘problem size’ is \( n/2^i \). To get down to the base case, we need this value to be 1 or less.

5 Example continued

- Depth of the leaves: \( \log_4 n \)
- Sum for each level: \( 3^i c(n/4^i)^2 = (3/16)^i cn^2 \).
- Sum across all levels:

6 A lopsided tree

\[ T(n) = T(n/3) + T(2n/3) + O(n) \]
7 Some branches terminate before others

Note that this recurrence does not produce a complete tree!

For \( n = 6 \) (assuming \( \lfloor n/3 \rfloor \) and \( \lfloor 2n/3 \rfloor \)):

\[
\begin{array}{c}
6c \\
2c & 4c \\
\Theta(1) & \Theta(1) & \Theta(1) \\
\Theta(1) & \Theta(1)
\end{array}
\]

Therefore, the sum from the previous slide gives an upper bound. We could also get a lower bound by truncating the tree at the level of its shallowest leaves.