csce750 — Analysis of Algorithms
Fall 2019 — Lecture Notes: Solving Recurrences — Recursion Trees

This document contains slides from the lecture, formatted to be suitable for printing or individual reading, and with some supplemental explanations added. It is intended as a supplement to, rather than a replacement for, the lectures themselves — you should not expect the notes to be self-contained or complete on their own.

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.

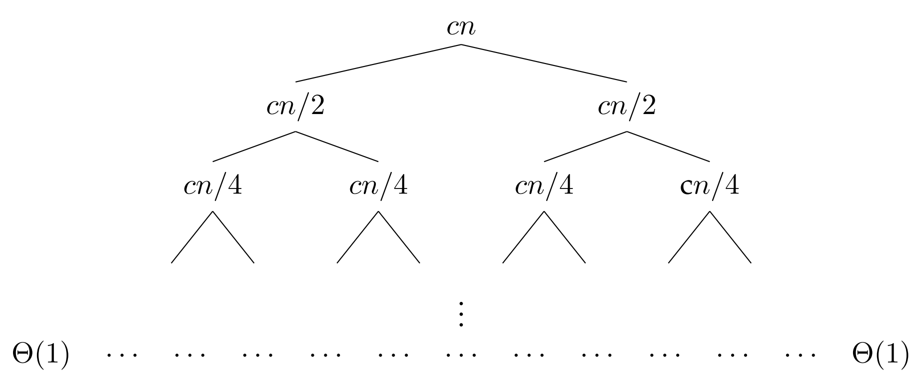
- **Children**: One for each appearance of a recurrent term.

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

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

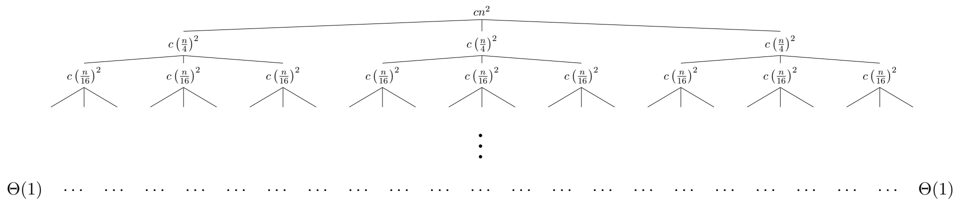


3 Example: Mergesort recurrence

- 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

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



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:

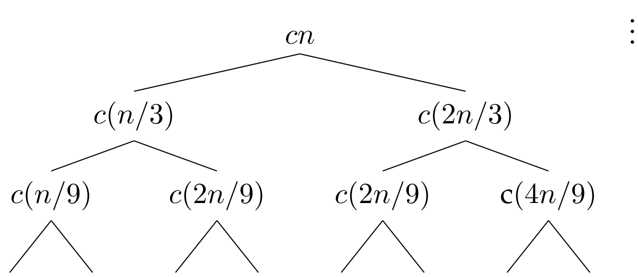
$$\begin{aligned} T(n) &= \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i cn^2 \\ &\leq \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 \\ &= \frac{1}{1-(3/16)} cn^2 \\ &= \frac{16}{13} cn^2 \\ &= O(n^2) \end{aligned}$$

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

> *At depth i, the 'problem size' is $n/2^i$. To get down to to the base case, we need this value to be 1 or less.*

6 A lopsided tree

$T(n) = T(n/3) + T(2n/3) + O(n)$

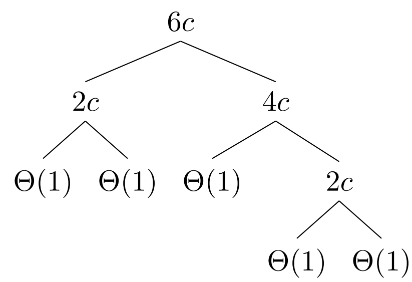


- Depth of the (deepest) leaves: $\log_{3/2} n$
- Sum for each level: $\leq cn$
- Sum across all levels: $cn \log_{3/2} n = O(n \log n)$.

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 $\lfloor 2n/3 \rfloor$):



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.