Master Method

Tufts University

Warm-up: substitution review

Given: $T(n) = 2T(\frac{n}{4}) + n$
Claim: $T(n) = O(n)$
I.e., for some c, n_0 we have $T(n) \le cn$ for all $n \ge n_0$.
PROOF: Base case: $T(1) = d$ for some constant d. For the claim to hold, we need $d \le c \cdot 1$. True if $c \ge d$.
Inductive step: Assume $T(k) \le ck$ for $1 \le k < n$. Then:

$$\begin{aligned} T(n) &= 2T(n/4) + n \\ &\le 2(c(n/4)) + n \\ &= cn/2 + n \\ &= cn - cn/2 + n \\ &= cn + (1 - c/2)n \\ &\le cn \text{ IF } 1 - \frac{c}{2} \le 0, \text{ or equivalently } 2 \le c \end{aligned}$$

Conclusion: For $c \ge \max\{d, 2\}$, the induction holds and we conclude $T(n) \le cn$ for all $n \ge 1$.

Solving Recurrences - The Big Picture

- Given: recursive relationship for runtime, e.g.

  $T(n) = 2T(\frac{n}{2}) + \Theta(n)$
  $T(n) = 2T(\frac{n}{4}) + \Theta(1)$
  $T(n) = 5T(\frac{2}{3}n) + O(n \log n)$
  $T(n) = T(\sqrt{n}) + O(1)$
  $T(n) = T(\frac{n}{2}) + 2T(\frac{n}{4}) + \Theta(\sqrt{n})$

- **Goal**: find closed asymptotic bound (i.e., $T(n) = \Theta(f(n))$.

- Methods:

  Substitution (Proof)
  Recursion tree (Proof or estimate)
  Master Method (Proof *if applicable)*

Recurrences: big picture

$T(n) = 2T(\frac{n}{2}) + n^2$ $\qquad$ $T(n) = 2T(\frac{n}{2}) + n$ $\qquad$ $T(n) = 2T(\frac{n}{2}) + 1$

For each recursion compute:

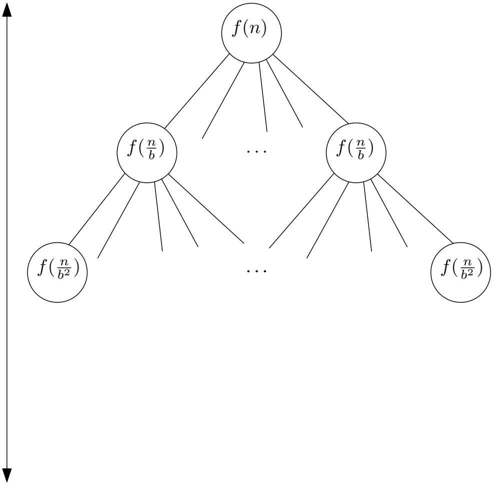
- The level sum for the first 3 levels and the leaf level
- A tight bound

Can we do this automatically?

Introducing Master Method

A formula for solving recurrences of the form:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$



Level 1 sum:

Level 2 sum:

# of leaves:

Height:

Summary

For $T(n) = aT(\frac{n}{b}) + f(n)$ the level sum is:

Lvl 1: $f(n)$

Lvl 2: $af(\frac{n}{b})$

Lvl 3: $a^2 f(\frac{n}{b^2})$

...

Leaf lvl: $O(1)n^{\log_b a}$

Top dominates If $f(n)$ is large
(i.e., $f(n) = n^{10}$, $a, b = 2$)

All levels equal Level 1 = Leaf level

Bottom dominates If $f(n)$ is large
(i.e., $a = 10$, $b = 2$ and $f(n) = n$)

Which one dominates?

Master method

$T(n) = aT(\frac{n}{b}) + f(n)$ solves to:

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } f(n) = O(n^{\log_b a - \varepsilon}) \\ \Theta(f(n) \log n) & \text{if } f(n) = \Theta(n^{\log_b a} \log^k n) \\ \Theta(f(n)) & \text{if } f(n) = \Omega(n^{\log_b a + \varepsilon}) \end{cases}$$

Under the following conditions:

- $a \geq 1$
- $b > 1$
- $f(n) > 0$ for $n > n_0$
- $k \geq 0$ (case 2)
- $af(\frac{n}{b}) < \delta f(n)$ for some $0 < \delta < 1$ (case 3)

Let's practice

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } f(n) = O(n^{\log_b a - \varepsilon}) \\ \Theta(f(n) \log n) & \text{if } f(n) = \Theta(n^{\log_b a} \log^k n) \\ \Theta(f(n)) & \text{if } f(n) = \Omega(n^{\log_b a + \varepsilon}) \end{cases}$$

- $T(n) = 4T(\frac{n}{2}) + n$

  $a = 4,\ b = 2 \rightarrow n^{\log_b a} = n^2$

  $f(n) = n$

  Case 1 applies ($f(n) = O(\#\text{ leaves}/n^{\varepsilon})$)

$\Rightarrow$ $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

Let's practice (II)

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } f(n) = O(n^{\log_b a-\varepsilon}) \\ \Theta(f(n)\log n) & \text{if } f(n) = \Theta(n^{\log_b a}\log^k n) \\ \Theta(f(n)) & \text{if } f(n) = \Omega(n^{\log_b a+\varepsilon}) \end{cases}$$

- $T(n) = 8T(\frac{n}{8}) + \Theta(n \log n)$

  $a = 8,\ b = 8 \rightarrow n^{\log_b a} = n$

  $f(n) = \Theta(n \log n)$

  Case 2 applies with $k = 1$ $(f(n) = \Theta(\#\text{ leaves}\cdot \log n))$

  $\Rightarrow$ $T(n) = \Theta(f(n)\log n) = \Theta(n \log^2 n)$

Let's practice (III)

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } f(n) = O(n^{\log_b a - \varepsilon}) \\ \Theta(f(n) \log n) & \text{if } f(n) = \Theta(n^{\log_b a} \log^k n) \\ \Theta(f(n)) & \text{if } f(n) = \Omega(n^{\log_b a + \varepsilon}) \end{cases}$$

- $T(n) = 9T(\frac{n}{3}) + \frac{n^2}{\log n}$

  $a = 9,\ b = 3 \to n^{\log_b a} = n^2$

  $f(n) = \frac{n^2}{\log n}$

  Master theorem does not apply

  $\Rightarrow$ Use substitution or recursion tree

Let's practice (IV)

- $T(n) = 2T(\frac{n}{2}) + \Theta(n)$
- $T(n) = 2T(\frac{n}{4}) + \Theta(1)$
- $T(n) = 5T(\frac{2}{3}n) + O(n \log n)$
- $T(n) = 3T(\frac{n}{9}) + \Theta(\sqrt{n} \log n)$
- $T(n) = T(\sqrt{n}) + O(1)$
- $T(n) = T(\frac{n}{2}) + 2T(\frac{n}{4}) + \Theta(\sqrt{n})$

Exercise!

Summary

- Master method is your new best friend
  - ☺ Easy to use
  - ☺ Fast
  - ☺ Both upper and lower bounds!
  - ☹ Does not always apply

Additional practice questions:

Let $T(n) = 3T(n/4) + f(n)$. Pick $f(n)$ such that:

- Case 1 applies
- Case 2 applies (with $k = 23$)
- Case 3 applies
- Master theorem does not apply

Can we use Master theorem in the following cases? why?

- $T(n) = 3T(2n) + 1$?
- $T(n) = T(n/2) + T(n/4) + n$?
- $T(n) = T(\sqrt{n}) + O(1)$?

Solve 100 random recurrences using master method