CMPT307: Recurrences

Week 3-3

Xian Qiu

Simon Fraser University

xianq@sfu.ca

Recurrences

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \le n_0, \\ aT(n/b) + f(n) & \text{otherwise,} \end{cases} \qquad a \ge 1, b > 1$$

- boundary condition: $T(n) = \Theta(1)$ if $n \le n_0$
- if boundary is clear, simplify it as $T(n) = aT(n/b) + f(n)$
- interpret n/b as either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$
- $T(n) = ?$

Outline

- substitution method
- recursion tree method
- master method

Substitution Method

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

guess $T(n) = O(n \log n)$, to show $T(n) \leq c \cdot n \log n$ for $n \geq n_0$

- let $c \geq T(2)$ and claim holds for $n = 2$
- assume $T(m) \leq c \cdot m \log m$ for all $2 \leq m < n$

$$\begin{aligned} T(n) &= 2T(n/2) + n \leq 2c \cdot \frac{n}{2} \log \frac{n}{2} + n \\ &= c \cdot n \log n - cn + n \leq c \cdot n \log n \qquad \text{let } c \geq 1 \end{aligned}$$

note: asymptotic notations cannot be used in inductive proof!

- now our guess is $T(n) = O(n)$
- by induction, $T(n) = 2O(n/2) + n = O(n)$ wrong!

  asymptotic notations do not have equivalence relation

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$

guess $T(n) = O(n)$ and to show $T(n) \leq cn$ for $n \geq n_0$

- it holds for $n = 1$ by letting $c \geq T(1)$
- assume it is true for $1 \leq m < n$ and consider n

$$T(n) \leq c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 = cn + 1$$

show $T(n) \leq cn - d$ instead

- assume true for $1 \leq m < n$ and consider n

$$\begin{aligned} T(n) &\leq (c\lfloor n/2 \rfloor - d) + (c\lceil n/2 \rceil - d) + 1 \\ &= cn - d - d + 1 \leq cn - d \end{aligned}$$

Changing Variables

$$T(n) = 2T(\sqrt{n}) + \log n$$

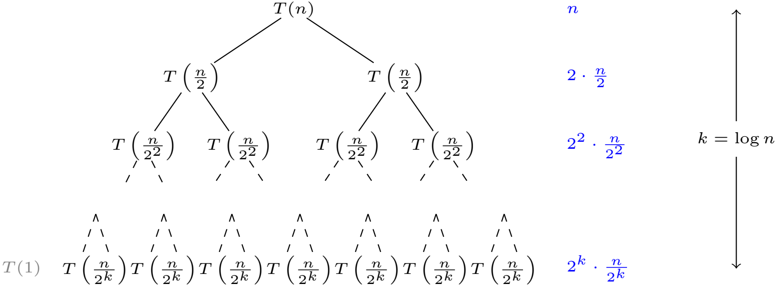
let $m = \log n$ and $S(m) := T(n)$, then

$$S(m) = T(2^m) \text{ and } S(m/2) = T(2^{m/2})$$

$$\begin{aligned} & T(2^m) = 2T(2^{m/2}) + m \\ \Rightarrow \quad & S(m) = 2S(m/2) + m \\ \Rightarrow \quad & S(m) = m \log m \\ \Rightarrow \quad & T(n) = S(m) = \log n \log \log n \end{aligned}$$

Recursion Tree Method

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



$$T(n) = 2^k T(1) + n \cdot k = nT(1) + n \log n = O(n \log n)$$

$$T(n) = 3T(n/4) + \Theta(n^2)$$

- ▷ tree height $k = \log_4 n$ and # leaves $= 3^k$
- ▷ additive terms:

$$\begin{aligned}
& cn^2 \\
& + 3c \cdot \left(\frac{n}{4}\right)^2 \\
& + 3^2 c \cdot \left(\frac{n}{4^2}\right)^2 \\
& + \cdots \\
& + 3^k c \cdot \left(\frac{n}{4^k}\right)^2 \;=\; \frac{1-(3/16)^k}{1-3/16} cn^2
\end{aligned}$$

$$\begin{aligned}
T(n) &= 3^k T(1) + \frac{1-(3/16)^k}{1-3/16} cn^2 = 3^{\log_4 n} T(1) + \frac{1-(3/16)^{\log_4 n}}{1-3/16} cn^2 \\
&= \Theta(n^{\log_4 3}) + O(n^2) = O(n^2)
\end{aligned}$$

Master Method

consider the recursive formula: $a \geq 1$, $b > 1$ are constants

$$T(n) = aT(n/b) + f(n)$$

Master theorem

1. if $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a})$
2. if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$
3. if $f(n) = \Omega(n^{\log_b a + \epsilon})$ and $af(n/b) \leq cf(n)$ for $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$

- compare $f(n)$ with $n^{\log_b a}$ and take the (strictly) larger one
- if they are the same, multiply a factor $\log n$
- $n^{\log_b a - \epsilon}$ polynomially smaller than $n^{\log_b a}$
- there are gaps, *e.g.* $f(n) = \Theta(n^{\log_b a} / \log n)$ and master theorem cannot be applied

Examples

$T(n) = 9T(n/3) + n$

- $f(n) = n$; $a = 9$, $b = 3$, $n^{\log_b a} = n^{\log_3 9} = n^2$
- case 1 applies, $T(n) = \Theta(n^2)$

$T(n) = T(2n/3) + 1$

- $f(n) = 1$; $a = 1$, $b = 3/2$, $n^{\log_b a} = 1$;
- case 2 applies, $T(n) = \Theta(\log n)$

$T(n) = 3T(n/4) + n \log n$

- $f(n) = n \log n$; $a = 3$, $b = 4$, $n^{\log_b a} = n^{\log_4 3}$
- case 3 may apply: to verify

$$af(n/b) \leq cf(n) \text{ for some } c < 1 \text{ and all large } n$$

- $3\frac{n}{4} \log \frac{n}{4} = \frac{3}{4}(n \log n - 2) \leq \frac{3}{4} n \log n$

$T(n) = 2T(n/2) + n \log n$ does not apply!