Recurrences

Useful for analyzing recursive algorithms

Describes a function in terms of its value on smaller inputs

For example, MergeSort

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 2T(n/2) + \Theta(1) + \Theta(n) & \text{if } n > 1 \end{cases}$$

$T(n) = \Omega(n \lg n)$

Solution Techniques

________________________ Guess bound then use mathematical induction to prove

_____________________ Convert recurrence to summation and bound summation

_________________ Simple solutions to recurrences of the form

$$T(n) = aT(n/b) + f(n), \quad a \geq 1, \; b > 1$$

In general, omit floors, ceilings and boundary conditions – they typically add only a constant factor.

Solution Techniques

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

Upper Bound: O

1. **Guess:** ____________________
2. **Show:** ___________________

Solution Techniques

Assume: T(n/2) $\leq$ c n/2 lg n/2

$$\begin{aligned} T(n) &\leq 2(c(n/2)\lg n/2) + n \\ &\leq cn \lg n/2 + n \\ &\leq cn \lg n - cn \lg 2 + n \\ &\leq cn \lg n - cn + n \end{aligned}$$

Want T(n) $\leq$ cn lg n.
To accomplish this we want (- cn lg 2 + n) to be $\leq$ 0.
Thus -cn + n $\leq$ 0, n $\leq$ cn, 1 $\leq$ c.
T(n) $\leq$ cn lg n, for c $\geq$ 1

Now for the boundary conditions ...

Assuming T(1) = 1, T(1) $\leq$ c1 lg 1 = 0, we cannot choose c large enough.

However, we are not constrained to show for ______, but ______________.

Extend boundary conditions:

```
T(1) = 1                  ; Recurrence no longer bottoms out
                          ; at T(1), but at T(2) and T(3)
T(2) = 2T(1) + 2 = 4
T(3) = 2T(1) + 3 = 5
T(2) <= c 2 lg 2 = 2c,    c >= 2
T(3) <= c 3 lg 3 = 4.75c, c >= 2
```

Making a Good Guess

1. Guess similar solutions to similar recurrences.

$$\mathrm{T(n) = 2T(n/2 + 42) + n}$$

Guess: ______________________

2. Narrow in on solution using loose upper and lower bounds.

$$\Omega(n) \longrightarrow \Theta(nlgn) \longleftarrow \mathrm{O}(n^2)$$

Problems: Lower-order terms may defeat mathematical induction of substitution method.

Example:

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

Guess: ______________

Show: _____________

Assume: __________________

$$T(n) \leq 2cn/2 + 1$$
$$= cn + 1 \not\leq cn$$

But, we cannot remove the +1.

Now let's try subtracting a lower-order term:

We know cn - b $\in$ O(n).

Guess: ______________

Show: __________________

Assume: _______________________

$$T(n) = 2T(n/2) + 1$$
$$T(n) \leq 2(cn/2 - b) + 1$$
$$= cn - 2b + 1 \leq cn - b$$
$$\text{True if } b - 1 \geq 0,\ b \geq 1 \text{ (okay)}$$

Changing Variables

Example: T(n) = 2T($\sqrt{n}$) + lg n

$$\text{Let } m = lgn, \;\; 2^m = n, \;\; n^{1/2} \;=\; (2^m)^{1/2} \;=\; 2^{m/2}$$
$$T(2^m) \;=\; 2T(2^{m/2}) \;+\; m$$

$$\text{Let } S(m) \;=\; T(2^m) \;=\; T(n)$$
$$S(m) \;=\; 2S(m/2) + m$$
$$S(m) \;=\; O(mlgm)$$
$$T(n) \;=\; O(mlgm) \;=\; O(lgn\; lglgn)$$

Example: Factorial

Factorial(n)
 if n < 1
 then return 1
 else return n * Factorial(n-1)

$$\text{T(n)} = \begin{cases} \Theta(1) & \text{if } n = 0 \\ T(n-1) \;+\; \Theta(1) & \text{if } n > 0 \end{cases}$$

Solution Techniques

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 0 \\ T(n-1) + \Theta(1) & \text{if } n > 0 \end{cases}$$

Guess: ______

Show: ______________

Assume: ______________________

$$T(n) \leq c(n\text{-}1) + \Theta(1)$$
$$= cn - c + \Theta(1)$$
$$\leq cn$$

If c $\geq \Theta(1)$. True for large enough c.

Initial Conditions: $T(1) = \Theta(1) \leq cn$

Example: Fibonacci

Fibonacci(n)
   if n < 2
   then return n
   else return Fibonacci(n-1) + Fibonacci(n-2)

$$T(n) = \begin{cases} 1 & \text{if } n < 2 \\ T(n-1) + T(n-2) + \Theta(1) & \text{if } n >= 2 \end{cases}$$

Solution Techniques

Guess: $T(n) = O(2^n)$, which means $T(n) \leq c2^n$

$$T(n) = T(n-1) + T(n-2) + \Theta(1)$$

$$<= \ c2^{n-1} \ + \ c2^{n-2} \ + \ \Theta(1)$$

$$= \ 1/2c2^n + 1/4c2^n + \Theta(1)$$

$$= \ 3/4c2^n + \Theta(1)$$

$$<= \ c2^n$$

If $1/4c2^n \geq \Theta(1)$ or $c \geq 4\Theta(1)/2^n$.

True for sufficiently large n.

Boundary: $T(0) = 1 \leq c2^0 = c$, $c \geq 1$

Actually, $T(n) = \Theta((\frac{1+\sqrt{5}}{2})^n)$.

Solution Techniques

Example:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n > 1 \end{cases}$$

```
T(n) = n + 2T(n/2)
     = n + 2[n/2 + 2T(n/4)]
     = n + n + 4T(n/4)
     = n + n + 4[n/4 + 2T(n/8)]
     = n + n + n + 8T(n/8)
     = n + n + n + 8[n/8 + 2T(n/16)]
     = n + n + n + n + 16T(n/16)
```

Note

The ith term of the series (i starts at 0) is $2^i T(n/2^i)$. The series ends when we hit T(1), or $n/2^i = 1 \longrightarrow n = 2^i \longrightarrow i = lgn$.

$$
\begin{aligned}
\mathrm{T(n)} &= \mathrm{n + n + n + ... +} 2^{lgn}T(1) \\
&= \mathrm{n + n + n + ... + nT(1)} \\
&= \Sigma_{i=0}^{lgn-1} n + \mathrm{nT(1)} \\
&= \mathrm{nlgn + n}\Theta(1) \\
&= \mathrm{nlgn} + \Theta(n) \\
&= \mathrm{nlgn} + o(nlgn) \\
&= \Theta(nlgn)
\end{aligned}
$$

Do not simplify asymptotic expressions until there is no summation in the expression.

Example: Factorial

Example - Wrong Way

$$
\begin{aligned}
\mathrm{T(n)} &= \mathrm{T(n\text{-}1)} + \Theta(1) \\
&= \Theta(1) + [\Theta(1) + \mathrm{T(n\text{-}2)}] \\
&= \Theta(1) + \mathrm{T(n\text{-}2)} \\
&= \Theta(1) + [\Theta(1) + \mathrm{T(n\text{-}3)}] \\
&= \Theta(1) + \mathrm{T(n\text{-}3)} \\
&= \Theta(1)
\end{aligned}
$$

Example - Right Way

$$
\begin{gathered}
\text{T(n)} = \text{T(n-1)} + \Theta(1) \\
= \Theta(1) + [\Theta(1) + \text{T(n-2)}] \\
= \Theta(1) + \Theta(1) + [\Theta(1) + \text{T(n-3)}] \\
\text{Terminates when n - i = 0, or i = n.} \\
= \Sigma_{i=0}^{n-1} \Theta(1) \ + \ T(0) \\
= \Theta(\Sigma_{i=0}^{n-1} 1) + \text{T}(0) \\
= \Theta(n) + \text{T}(0)
\end{gathered}
$$

Example - Wrong Way

$$
T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(n/3) + \Theta(n) & \text{if } n > 1 \end{cases}
$$

$$
\begin{gathered}
\text{T(n)} = \Theta(n) + \text{T(n/3)} \\
= \Theta(n) + [\Theta(n/3) + \text{T}(\text{n}/3^2)] \\
= \Theta(n) + \Theta(n) + \text{T}(\text{n}/3^2) \\
\text{Terminates when } n/3^i = 1\text{, or } i = log_3 n. \\
= \Sigma_{i=0}^{log_3 n-1} \Theta(n) + \text{T}(1) \\
= \Theta(n log_3 n)\text{, not what we want!!!}
\end{gathered}
$$

Example - Right Way

$$
\begin{gathered}
\text{T(n)} = \Theta(n) + \text{T(n/3)} \\
= \Theta(n) + [\Theta(n/3) + \text{T}(\text{n}/3^2)] \\
= \Theta(n) + \Theta(n/3) + \text{T}(\text{n}/3^2) \\
= \Theta(n) + \Theta(n/3) + \Theta(n/3^2) + \text{T}(\text{n}/3^3)
\end{gathered}
$$

$$= \Sigma_{i=0}^{log_3 n-1} \Theta(n/3^i) + \mathrm{T}(1)$$
$$= \Theta(n\, \Sigma_{i=0}^{log_3 n-1} (1/3)^i) + \mathrm{T}(1)$$
$$= \Theta(n(\frac{(1/3)^{log_3 n}-1}{1/3-1})) + \mathrm{T}(1)$$
$$= \Theta(3/2n(1-1/n)) + \mathrm{T}(1)$$
$$= \Theta(\frac{3n}{2} - \frac{3n}{n}) + \mathrm{T}(1)$$
$$= \Theta(n)$$

Solution Techniques

For example, Merge Sort fits in this format with a=2, b=2, f(n) = $\Theta(n)$

Master Theorem

For the recurrence relation ______________________________ with a $\geq$ 1, b $>$ 1 and T(n) on non-negative integers, T(n) can be asymptotically bounded as follows:

1. If f(n) = O($n^{log_b a\, -\, \epsilon}$) for some constant $\epsilon > 0$, then T(n) = $\Theta(n^{log_b a})$.

2. If f(n) = $\Theta(n^{log_b a})$, then T(n) = $\Theta(n^{log_b a} lgn)$.

3. If f(n) = $\Omega(n^{log_b a\, +\, \epsilon})$ for some constant $\epsilon > 0$, and if af(n/b) $\leq$ cf(n) for some constant c $<$ 1 and large n, then T(n) = $\Theta(f(n))$.

Which is larger, f(n) or $n^{log_b a}$?
By a factor of n^{ϵ}, or polynomially larger...

Example

$$T(n) = 2T(n/2) + n$$
$$a=2,\ b=2,\ f(n) = n$$
$$n^{log_2 2} = n = \Theta(n), \text{ Case 2}$$
$$T(n) = \Theta(nlgn)$$

Example

$$T(n) = 9T(n/3) + n$$
$$a=9,\ b=3,\ f(n) = n$$
$$n^{log_3 9} = n^2 = \Theta(n^2)$$
$$f(n) = n = O(n^{log_3 9-\epsilon}), \text{ where } \epsilon = 1, \text{ Case 1}$$
$$T(n) = \Theta(n^2)$$

Example

$$T(n) = 3T(n/4) + nlgn$$
$$a=3,\ b=4,\ f(n) = nlgn$$
$$n^{log_4 3} = O(n^{0.793})$$
$$f(n) = \Omega(n^{log_4 3+\epsilon}),\ \epsilon \approx 0.2, \text{ Case 3}$$
$$\text{For large n, } af(n/b) = 3(n/4)lg(n/4) \leq 3/4nlgn = cf(n),\ c=3/4$$
$$T(n) = \Theta(nlgn)$$

Example: Use all three methods

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

Guess?

Master

$$\text{a=2, b=2, } log_b a = 1, \text{ f(n)} = n^3$$
$$\text{f(n)} = \Omega(n^{1+2}) = \Omega(n^3), \epsilon = 2$$
$$\text{2f(n/2)} \leq \text{cf(n) for c<1?}$$
$$2 \ (n/2)^3 \leq \text{c}n^3$$
$$1/4 \ n^3 \leq \text{c}n^3, \ 1/4 \leq \text{c} < 1, \text{ Case 3}$$
$$\text{T(n)} = \Theta(n^3)$$

Substitution

$$\text{T(n)} = \text{O}(n^3) \leq \text{c}n^3, \text{ Assume T(n/2)} \leq \text{c}(n/2)^3$$
$$\text{T(n)} \leq 2\text{c}(n/2)^3 + n^3$$
$$= 1/4\text{c}n^3 + n^3$$
$$= (1 + \text{c}/4)n^3$$
$$\leq \text{c}n^3 \text{ when}$$
$$1 + \text{c}/4 \leq \text{c}, \ 3/4\text{c} \geq 1, \ \text{c} \geq 4/3$$

$$\text{T(n)} = \Omega(n^3) \geq \text{c}n^3, \text{ T(n/2)} \geq \text{c}(n/2)^3$$
$$\text{T(n)} \geq 2\text{c}(n/2)^3 + n^3$$
$$= 1/4\text{c}n^3 + n^3$$

$= (1 + c/4)n^3$
$\geq cn^3$ when
$1 + c/4 \geq c$, $3/4c \leq 1$, $c \leq 4/3$

Thus, $T(n) = \Theta(n^3)$.

Iteration

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

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

$$= \ n^3 \ + \ \frac{n^3}{2^2} \ + \ 4T(n/4)$$

$$= \ n^3 \ + \ \frac{n^3}{2^2} \ + \ 4[(n/4)^3 \ + \ 2T(n/8)]$$

$$= \ n^3 \ + \ \frac{n^3}{2^2} \ + \ \frac{n^3}{4^2} \ + \ 8T(n/8)$$

These are terms i=0, i=1, i=2, i=3

$$= \ n^3 \ + \ \frac{n^3}{2^2} \ + \ \frac{n^3}{4^2} \ + \ \frac{n^3}{8^2} \ + \ ... \ + \ 2^i T(\frac{n}{2^i})$$

Terminates when $n/2^i = 2$, $T(2) = \Theta(1)$.
$n = 2^{i+1}$, $lgn = i+1$, $i = lgn - 1$

$$= \ \sum_{i=0}^{lgn-2} \frac{n^3}{(2^i)^2} \ + \ 2^{lgn-1} T(2)$$

$$= \ n^3 \sum_{i=0}^{lgn-2} (1/4)^i \ + \ 1/2n\Theta(1)$$

$$= \ n^3(\frac{(1/4)^{lgn-1} - 1}{1/4 - 1}) \ + \ 1/2\Theta(n)$$

$$= \ n^3(\frac{1 - (2^{-2})^{lgn-1}}{3/4}) \ + \ \Theta(n)$$

Note that $(1/4)^{lgn-1} = 2^{-2lgn+2}$
$= 2^{lgn^{-2}} \ * \ 4 = 4n^{-2}$

$$= \ 4/3n^3(1 - 4n^{-2}) \ + \ \Theta(n)$$

$$= \ 4/3n^3 - 16/3n + \Theta(n)$$

$$= \ \Theta(n^3)$$
