Lecture 4, Oct. 1, 2009

Initial Conditions

- Can initial conditions affect the solution ?

$$T(n) = \left[T(n/2)^2 \right]$$

$$T(1) = 2 \;\Rightarrow\; T(n) = 2^n$$
$$T(1) = 3 \;\Rightarrow\; T(n) = 3^n$$
$$T(1) = 1 \;\Rightarrow\; T(n) = 1$$

- n was assumed to be a power of 2.

39

Iterating recurrences

- Example: $T(n) = 4T(n/2) + n = n + 4T(n/2)$

$$= n + 4(n/2 + 4T(n/4)) = n + 2n + 16T(n/4)$$
$$= n + 2n + 16[n/4 + 4T(n/8)] = n + 2n + 4n + 64T(n/8)$$
$$= n + 2n + 4n + 8n + \cdots = \underbrace{n\sum_{i=0}^{\lg n - 1} 2^k}_{n\frac{2^{\lg n}-1}{2-1}} + \underbrace{4^{\lg n} T(1)}$$

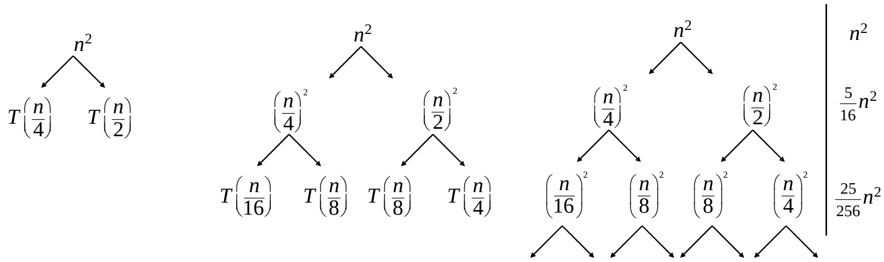
$\Theta(n^2)$ $\Theta(n^2)$

- Disadvantages:
  - » Tedious
  - » Error-prone
- Use to generate initial guess, and then prove by induction !

40

Recursion Tree

- Example: $T(n) = T(n/4) + T(n/2) + n^2$



- At *k*-th level we get a general formula: *i* steps right, *k-i* left

$$n^2 \sum_i \binom{k}{i} \left[2^{-i} 4^{-(k-i)}\right]^2 = n^2 \sum_i \binom{k}{i} \left[4^{-i} 16^{-(k-i)}\right] =$$

$$= n^2 \left[\frac{1}{4} + \frac{1}{16}\right]^k = n^2 \left[\frac{5}{16}\right]^k$$

- Summing over all k, geometric sum, sums to $\Theta(n^2)$ (overcount, since T(1)=1)

41

Master Method

- Consider the following recurrence: $T(n) = aT(n/b) + f(n);\ a \geq 1, b > 1$

1. $f(n) = O(n^{\lg_b a - \varepsilon}), \varepsilon > 0 \quad \Rightarrow \quad T(n) = \Theta(n^{\lg_b a})$
2. $f(n) = \Theta(n^{\lg_b a} \lg^k n),\ \ k \geq 0 \quad \Rightarrow \quad T(n) = \Theta(n^{\lg_b a} \lg^{k+1} n)$
3. $\left.\begin{array}{l} f(n) = \Omega(n^{\lg_b a + \varepsilon}), \varepsilon > 0 \\ af(n/b) \leq cf(n) \text{ for some } c < 1 \end{array}\right\} \Rightarrow \quad T(n) = \Theta(f(n))$

- Let $Q = n^{\lg_b a}$. Then the cases are:
  - » f polynomially smaller than Q.
  - » f is larger than Q by a polylog factor.
  - » f polynomially larger than Q.

42

Examples

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

$n^{\lg_b a} = n^{\lg_2 2} = n$

$\Theta(n)/n = \Theta(1) = \Theta(\lg^0 n) \Rightarrow$ case 2 $\Rightarrow T(n) = \Theta(n \lg n)$

Strassen's matrix multiplication

$T(n) = 7T(n/2) + \Theta(n^2)$

$n^{\lg_b a} = n^{\lg_2 7}$

$\frac{\Theta(n^2)}{n^{\lg_2 7}} \approx O(n^{-0.8}) \Rightarrow$ case 1 $\Rightarrow T(n) = \Theta(n^{\lg_2 7})$ ← Better than n^3 !!!

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

$\frac{n^3}{n^{\lg_2 4}} = n \Rightarrow$ case 3 $\Rightarrow T(n) = \Theta(n^3)$

(*Note*: need to check the additional condition $\exists 0 < c < 1$ s.t. $4(n/2)^3 \le cn^3$)

Does the method always apply ?

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

$$\frac{n^2/\lg n}{n^{\lg_2 4}} = \frac{1}{\lg n} \neq \begin{cases} O(n^{-\varepsilon}), \varepsilon > 0 \\ \Theta(\lg^k n), k \ge 0 \\ \Omega(n^{\varepsilon}), \varepsilon > 0 \end{cases}$$

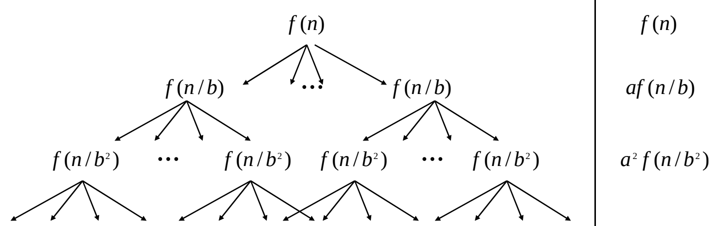
Upper bound: $4T(n/2) + n^2 \Rightarrow$ case 2 $\Rightarrow \Theta(n^2 \lg n)$

Lower bound: $4T(n/2) + n^{2-\varepsilon} \Rightarrow$ case 1 $\Rightarrow \Theta(n^2)$

Exact Answer: $\Theta(n^2 \lg\lg n)$

Build recursion tree

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



Last row: $\Theta(a^{\lg_b n}) = \Theta(n^{\lg_b a})$ elements, each one $\Theta(1)$.

Total: $\Theta(n^{\lg_b a}) + \sum_{i=0}^{\lg_b n-1} a^i f(n/b^i)$

Which term dominates ?

First case: "f(n) small"

Total: $\Theta(n^{\lg_b a}) + \sum_{j=0}^{\lg_b n-1} a^j f(n/b^j)$

$\frac{n^{\lg_b a}}{f(n)} = \Omega(n^{\varepsilon}) \Rightarrow \exists c$ s.t for "large enough n", $f(n) \leq cn^{\lg_b a}/n^{\varepsilon}$

$a^j f(n/b^j) \leq ca^j (n/b^j)^{\lg_b a-\varepsilon} = cn^{\lg_b a-\varepsilon} a^j \frac{b^{j\varepsilon}}{b^{j\lg_b a}} = cn^{\lg_b a-\varepsilon} b^{j\varepsilon}$

The ratio summed over all possible j: $\frac{b^{\varepsilon \lg_b n}-1}{b^{\varepsilon}-1} = \Theta(n^{\varepsilon})$.

Total: $O(n^{\lg_b a})$.

Lower bound is trivial
(Why ?? Because first term in the original expression was already $\Theta(n^{\lg_b a})$.)

Second case

$f(n) = \Theta(n^{\lg_b a} \lg^k n)$ Total: $T(n) = \Theta(n^{\lg_b a}) + \sum_{j=0}^{\lg_b n - 1} a^j f(n/b^j)$

$\sum \underbrace{a^j \left(\frac{n}{b^j}\right)^{\lg_b a}}_{=n^{\lg_b a}} \underbrace{\lg^k \left(\frac{n}{b^j}\right)}_{\le \lg^k n} = O(\lg^{k+1} n) n^{\lg_b a}$ (there are $O(\lg n)$ elements in the sum)

This is an UPPER bound ! How to prove the lower bound ??

Rough and easy approach:

$$\sum_{j=1}^{\lg_b n - 1} \lg^k \left(\frac{n}{b^j}\right) \ge \sum_{j=1}^{(\lg_b n)/2} \lg^k \left(\frac{n}{b^j}\right) \ge \sum_{j=1}^{(\lg_b n)/2} \lg^k \sqrt{n} = (\text{const}) \cdot \lg^{k+1} n$$

(Note that we use the assumption that $k \ge 0$)
(Formally explain how to deal with (log n)/2 being non-integer)

47

Total: $\Theta(n^{\lg_b a}) + \sum_{i=0}^{\lg_b n - 1} a^i f(n/b^i)$

Third case

$a^j f(n/b^j) \le c^j f(n)$ for some $c < 1$, and $f(n) = \Omega(n^{\lg_b a + \varepsilon})$

$$\Rightarrow \sum_{i=0}^{\lg_b n - 1} c^j f(n) = \Theta(f(n))$$

$$\Rightarrow \sum_{i=0}^{\lg_b n - 1} a^j f(n/b^j) = O(f(n))$$ Note Big-Oh and not Theta !

The first term is $\Theta(n^{\lg_b a}) = O(f(n))$

TOTAL: $\Theta(f(n))$

Why Theta and not plain big-O ?

48