Agenda

We've done

- Growth of functions
- Asymptotic Notations ($O, o, \Omega, \omega, \Theta$)

Now

- Recurrence relations, solving them, Master theorem

Examples of recurrence relations

FibA

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

Binary search

$$T(n) \leq T(\lceil n/2 \rceil) + \Theta(1)$$

Merge sort

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) \tag{1}$$

and many others

$$\begin{aligned} T(n) &= 4T(n/2) + n^2 \lg n \\ T(n) &= 3T(n/4) + \lg n \\ T(n) &= T(n/a) + T(a) \end{aligned}$$

Recall the way to interpret (1): "$T(n)$ is $T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$ plus some function $f(n)$ which is $\Theta(n)$"

Methods of solving recurrent relations

- Guess and induct
- Master Theorem
- Generating functions and many others

Guess and induct

- Guess a solution
  - Guess by substitution
  - Guess by recurrence tree
- Use induction to show that the guess is correct

Guess by substitution - Example 1

Example (The FibA algorithm)

$$T(n) = \begin{cases} a & \text{if } n \leq 1 \\ T(n-1) + T(n-2) + b & \text{if } n \geq 2 \end{cases}$$

Guess by iterating the recurrence a few times:

- $T(0) = a$, $T(1) = a$
- $T(2) = 2a + 1b$
- $T(3) = 3a + 2b$
- $T(4) = 5a + 4b$
- $T(5) = 8a + 7b$
- ...

So, what's $T(n)$?

Guess by substitution - Example 1

The guess

$$T(n) = (a+b)F_{n+1} - b \tag{2}$$

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n = \Theta(\phi^n), \tag{3}$$

where F_n is the nth Fibonacci number, ϕ is the golden ratio
Conclude with

$$T(n) = \Theta(\phi^n) \tag{4}$$

We can show (2), (3) & (4) by induction.

Guess by substitution – Example 2

Example (Merge Sort)

$$\begin{aligned} T(1) &= \Theta(1) \\ T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) \end{aligned}$$

Clean up the recurrence before guessing

It is often safe to ignore the issue of integrality:

$$T(n) \approx T(n/2) + T(n/2) + cn = 2T(n/2) + cn.$$

Guess by substitution – Example 2

$$\begin{aligned} T(n) &= 2T(n/2) + cn \\ &= 2\left(2T(n/4) + c\frac{n}{2}\right) + cn \\ &= 4T(n/4) + 2cn \\ &= 4\left(2T(n/8) + c\frac{n}{4}\right) + 2cn \\ &= 8T(n/8) + 3cn \\ &= \dots \\ &= 2^k T(n/2^k) + kcn \\ &= \dots \\ &= 2^{\lg n} T(n/2^{\lg n}) + cn \lg n \\ &= \Theta(n \lg n) \end{aligned}$$

Guess by substitution – Example 2

- Rigorously, we have

$$\begin{aligned} T(1) &= c_0 \\ T(n) &\geq T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + c_1 n \\ T(n) &\leq T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + c_2 n \end{aligned}$$

- Guess: $T(n) = \Theta(n \lg n)$.
- By induction, show that there are constants $a, b > 0$ such that

$$an \lg n \leq T(n) \leq bn \lg n.$$

Now try

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

"Cleaning up" before solving: Ignore annoying constants and integrality issue

- To (sort of) see why integrality isn't important, consider

$$T(n) = 2T(\lfloor n/2 \rfloor + 17) + n.$$

- Approximate this by ignoring both the integrality issue and the annoying constant 17

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

- The guess is then $T(n) = O(n \lg n)$. (You should prove it.)

Common mistake

$$T(n) \leq 2c\lfloor n/2 \rfloor + n \leq cn + n = O(n)$$

“Cleaning up” before solving: Change of variable

Solve

$$T(n) = 2T(\sqrt{n}) + 1$$

Let $m = \lg n$, then

$$T(2^m) = 2T(2^{m/2}) + 1$$

Let $S(m) = T(2^m)$, then

$$S(m) = 2S(m/2) + 1.$$

Hence,

$$S(m) = O(m).$$

Thus,

$$T(n) = S(\lg n) = O(\lg n).$$

Guess by recurrence tree – Example 1

Example

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

Recursion tree suggests $T(n) = O(n^2)$. Prove rigorously by induction.

Example (Now try this)

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

Master Theorem

Let $a \geq 1$, $b > 1$ be constants. Suppose

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

where n/b could either be $\lceil n/b \rceil$ or $\lfloor n/b \rfloor$. Then

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

Examples and Notes

Examples

- $T(n) = 8T(n/2) + n^5$
- $T(n) = 8T(n/2) + n^4$
- $T(n) = 8T(n/2) + n^3$
- $T(n) = 3T(n/2) + n^2$
- $T(n) = 3T(n/2) + n$

Notes

- There is a gap between case 1 & case 2
- There is a gap between case 2 & case 3
- There is a gap within case 3

Other methods of solving recurrences

- Generating functions
- Hypergeometric series
- Finite calculus, finite differences
- ...

Further readings

- "$A = B$," by M. Petkovsek, H. Wilf, D. Zeilberger
- "Concrete mathematics," R. Graham, D. Knuth, O. Patashnik
- "Enumerative combinatorics," R. Stanley (two volumes)
- "Theory of partitions," G. Andrews