Recurrences

A recurrence is an equation or inequality that relates a function's value in terms of its values on smaller inputs.

Used to analyze the runtime of recursive algo's.

MergeSort example

Input: List A of n numbers

Output: Same list, sorted in ascending order

MergeSort(A):

O(1) [if A has ≥2 items then]

$T(\frac{n}{2})$ Mergesort (1st half of A)

$T(\frac{n}{2})$ Mergesort (2nd half of A)

$\Theta(n)$ [Merge the 2 halves]

Let $T(n)$:= time to mergesort n items:

$$T(n) = 2T(\tfrac{n}{2}) + \underbrace{\Theta(n)}_{cn}$$

Solve for $T(n)$.

3 methods:

1. Substitution method — rigorously verify that a proposed solution is correct. Work not nec. good at finding a solution.
2. Recursion tree method — Map out the entire run of the algo as a tree (subtrees correspond to recursive calls), add up the nodes of tree.
3. Master Method — theorem providing solutions to common recurrences.

Example

$T(n)$ = time to mergesort n items

$$T(n) = 2T(\tfrac{n}{2}) + \underbrace{\Theta(n)}_{\text{can just use } n}$$

$$T(n) = 2T(\tfrac{n}{2}) + n$$

Prop: $T(n) = \Theta(n \lg n)$

Θ or O

Proof (substitution method):

[Essentially proof by strong induction]

$T(n) = O(n \lg n)$ (upper bound):

Assume the inequality

$T(m) \le c\, m \lg m$ (all $m < n$)

holds. WTS $T(n) \le cn \lg n$ (same c)

[value of c to be determined]

$$T(n) = 2T(\tfrac{n}{2}) + n$$

$$\le 2c(\tfrac{n}{2})\lg(\tfrac{n}{2}) + n$$

[induction hypothesis with $m = \frac{n}{2}$]

$$= cn\lg(\tfrac{n}{2}) + n$$

$$= cn\left(\lg n - \lg 2\right) + n$$

$$= cn(\lg n - 1) + n$$

$$= \underbrace{cn\lg n}_{\text{WTS}} - cn + n$$

WTS $\le cn \lg n$ provided $-cn + n \le 0$

$n \le cn$

$1 \le c$

Lower bound:

assume (ind. hypothesis) $T(m) \ge c' m \lg m$ (all $m < n$) for some $c' > 0$ to be determined

$$T(n) = 2T(\tfrac{n}{2}) + n$$

$$\ge 2c'(\tfrac{n}{2})\lg(\tfrac{n}{2}) + n$$

$$= c'n\lg(\tfrac{n}{2}) + n$$

$$= c'n(\lg n - 1) + n$$

$$= \underbrace{c'n\lg n}_{\text{WTS}} - c'n + n$$

WTS $\ge c'n \lg n$ provided $-c'n + n \ge 0$

$n \ge c'n$

$1 \ge c'$ ///

Detail 1: n may be odd, so $T(\frac{n}{2})$ makes no sense.

Actual recurrence for MergeSort is

$$T(n) = T\left(\lfloor \tfrac{n}{2} \rfloor\right) + T\left(\lceil \tfrac{n}{2} \rceil\right) + n$$

Prove that $T(n) = \Theta(n \lg n)$

U.B.: assume $T(n) \le c\, m \lg m$ $(m < n)$

$T(n) = T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + n$

$\le c\lfloor \frac{n}{2} \rfloor \lg \lfloor \frac{n}{2} \rfloor + T(\lceil \frac{n}{2} \rceil) + n$

$\le c(\frac{n}{2}) \lg(\frac{n}{2}) + c\lceil \frac{n}{2} \rceil \lg \lceil \frac{n}{2} \rceil + n$

[$x \in \mathbb{R}$, $x - 1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x + 1$]

$\le \frac{cn}{2}(\lg n - 1) + c(\frac{n}{2} + 1)\lg(\frac{n}{2} + 1) + n$

[lg function grows slowly]

$\le \frac{cn}{2}(\lg n - 1) + c(\frac{n}{2} + 1)\lg(\frac{3n}{4}) + n$

[because $\frac{n}{2} + 1 \le \frac{3}{4}n$ for all sufficiently large n]

$= \frac{cn \lg n}{2} - \frac{cn}{2} + \frac{cn}{2}\lg(\frac{3n}{4}) + c\lg(\frac{3n}{4}) + n$

$= \frac{cn \lg n}{2} + \frac{cn}{2}(\lg n - \lg(\frac{4}{3})) + c\lg(\frac{3n}{4}) + n$

$= cn \lg n - \frac{cn}{2}\lg(\frac{4}{3}) + c\lg(\frac{3n}{4}) + n$

$\le cn \lg n$ provided

$-\frac{cn}{2}\lg(\frac{4}{3}) + c\lg(\frac{3n}{4}) + n \le 0$

$0 \le n\left(\underbrace{\frac{c}{2}\lg(\frac{4}{3}) - 1}_{\text{make this } > 0}\right) - c\lg(\frac{3n}{4})$

true for all sufficiently large n (because $c\lg(\frac{3n}{4}) = o(n)$)

Just need $\frac{c}{2}\lg(\frac{4}{3}) > 1$

$c > \frac{2}{\lg(\frac{4}{3})}$ [& c big enough to handle the finitely many base cases]

LB has a similarly messy analysis.

For asymptotics when using induction, you can assume that n is sufficiently large.

Only worry about the inductive case. Ignore the base cases generally.

Ex:

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

Claim: $T(n) = \Theta(n^2)$

Proof: Assume $T(m) \le cm^2$ for $m < n$.

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

$= cn^2 + n$

$\le cn^2$ if $n \le 0$ — false! — doesn't work

Stronger inductive hypothesis:

$T(m) \le cm^2 - dm$ (c, d constants to be determined)

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

$\le 4\left(c(\frac{n}{2})^2 - d(\frac{n}{2})\right) + n$

$= cn^2 - 2dn + n$ ($-dn - dn$)

$\le cn^2 - dn$

provided $-dn + n \le 0$

$n \le dn$

$1 \le d$

[constrained by the base cases]

LB

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

$\ge 4(c(\frac{n}{2})^2) + n$

$= cn^2 + n$

$\rightarrow \ge cn^2$ (any $c > 0$ works for inductive case)