CSCE 750
9/5/2023

Solving recurrences

First: Approx sum by an integral

$$\sum_{i=\ell}^{u} f(i)$$

If f is monotone ascending:

$$\int_{\ell-1}^{u} f(x)\,dx \;\le\; \sum_{i=\ell}^{u} f(i) \;\le\; \int_{\ell}^{u+1} f(x)\,dx$$

assuming integrals finite

If monotone descending, ... $\ge$ ... $\ge$ ...

A recurrence is an equation equating a function $T(n)$ with $T(m)$ for values $m < n$.

A recurrence expresses the run time of a recursive algorithm.

$n = u - \ell + 1$

Ex: MergeSort($A[\ell \ldots u]$) // sort $A[\ell..u]$

if $\ell < u$ then

→ $B := \text{MergeSort}(A[\ell \ldots \frac{\ell+u}{2}])$

→ $C := \text{MergeSort}(A[\frac{\ell+u}{2}+1, \ldots, u])$

$\Theta(n)$ → return Merge(B, C)
endif

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

$$T(n) = \Theta(n) + 2T\left(\frac{n}{2}\right)$$

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

Techniques to solve a recurrence:

1. Substitution method
   proof of correctness (by induction)
2. Tree method
   map out entire tree of recursive calls, add it up.
3. Master Method
   simple formulas that work in lots of common cases.

Substitution method:

$$T(n) = 2T\left(\frac{n}{2}\right) + \overbrace{\Theta(n)}^{\text{can assume constant is 1}}$$

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

Guess that $T(n) = \underbrace{\Theta}_{O \text{ and } \Omega}(n \lg n)$

Upper bound: Let n be sufficiently large.

Assume (inductive hypothesis) that

$$T(m) \le c\, m \lg m \quad \text{for } m < n.$$

[c to be determined]

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

$\le 2c(\frac{n}{2}) \lg(\frac{n}{2}) + n$ [inductive hyp]

$= cn(\lg n - \underbrace{\lg 2}_{1}) + n$

$= cn \lg n - cn + n$

$= \underbrace{cn \lg n}_{\text{WTS}} - n(c-1)$

$\le cn \lg n$ provided $\underbrace{n(c-1) \ge 0}_{\text{true if } c \ge 1}$

$\therefore T(n) \le cn \lg n$ for any $c \ge 1$

False proof that $T(n) = O(n)$:

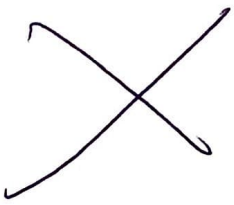
$$\frac{T(n) = O(n) \le cn}{T(n) = 2T(\frac{n}{2}) + n}$$

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

$= cn + n$

$= (c+1)n = O(n)$

bigger than n in the ind. hyp. — not really a constant!



Lower bound: Ind hyp: $T(m) \geq cm \lg m$ for $m < n$.

Proof:

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

$\geq 2c(\frac{n}{2})\lg(\frac{n}{2}) + n$

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

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

$\geq cn\lg n$

want

provided $(1-c)n \geq 0$ that is, $0 < c \leq 1$

Generally ignore the base cases — for upper bd, just pick c large enough so that the ind hyp is true for finitely many values of small m.

Recall (from day 1) to divide-and-conquer algos for multiplying two n-digit numbers.

1st Algo: 4 recursive calls on half-size inputs + $\Theta(n)$ of other stuff

time of 1st algo on 2 n-digit numbers → $T_1(n) = 4T_1(\frac{n}{2}) + n$] guess $T_1(n) = \Theta(n^2)$

2nd algo — same except 3 rec calls not 4.

$$T_2(n) = 3T_2(\frac{n}{2}) + n$$

$T_1(n) = O(n^2)$: Assume $T_1(m) \le cm^2$ for $m < n$

$$T_1(n) = 4T_1(\tfrac{n}{2}) + n$$

$$\le 4c(\tfrac{n}{2})^2 + n$$

$$= cn^2 + n$$

$$\le cn^2 \quad \text{impossible!}$$

Need a stronger inductive hypothesis:

upper bound

Assume $T_1(m) \leq cm^2 - dm \quad [m < n]$

$$T_1(n) = 4T_1\left(\frac{n}{2}\right) + n$$

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

$$= 4\left(\frac{cn^2}{4} - \frac{dn}{2}\right) + n$$

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

$$= cn^2 - (2d-1)n$$

$$\leq cn^2 - dn$$

need ↑

provided

$(2d-1)n \geq dn$

$2d - 1 \geq d$

$d \geq 1$

true for any $d \geq 1$.

Lower bound

Assume $T_1(m) \geq cm^2 \quad [m < n]$

$$T_1(n) = 4T_1\left(\frac{n}{2}\right) + n$$

$$\geq 4c\left(\frac{n}{2}\right)^2 + n$$

$$= cn^2 + n \geq cn^2 \quad \checkmark$$

$$T_2(n) = 3T_2\left(\frac{n}{2}\right) + n \Big] \quad T_2(n) = \Theta(n^{\lg 3})$$

Upper bound: Assume $T_2(m) \le cm^{\lg 3} - dm \quad (m < n)$

$$T_2(n) = 3T_2\left(\frac{n}{2}\right) + n$$

$$\le 3\left(c\left(\frac{n}{2}\right)^{\lg 3} - d\left(\frac{n}{2}\right)\right) + n$$

$$= \frac{3cn^{\lg 3}}{\underbrace{2^{\lg 3}}_{3}} - \frac{3dn}{2} + n$$

$$= cn^{\lg 3} - dn \underbrace{- \frac{dn}{2} + n}$$

$$\le cn^{\lg 3} - dn \quad \text{provided}$$

$$-\frac{dn}{2} + n \le 0$$

$$-\frac{d}{2} + 1 \le 0$$

$$1 \le \frac{d}{2}$$

$$2 \le d$$

Try the lower bound yourself.