Algo

Recurrence Relations II

2/5/19

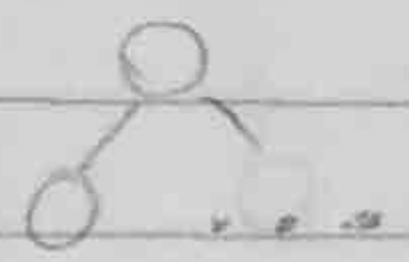
Specific Form

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

$(n)^2$ / \\ $(n/2)^2 \quad (n/2)^2$ / \\ / \\ $(\frac{n}{4})^2 \; (\frac{n}{4})^2 \; (\frac{n}{4})^2 \; (n/4)^2$	$1 + \log_2(n)$	$2^i (n/2^i)^2$

General Form

$T(n) = aT(n/b) + n^c$



	$\log_b(n) +$	$a^i (n/b^i)^c$

Comparing 3 recurrences

① $T(n) = 2T(n/2) + n$	sums	② $T(n) = T(n/2) + n$	sums
n	n	n	n
/ \\		$\vert$	
$n/2 \quad n/2$	$2(n/2)$	$n/2$	$n/2$
/ \\ / \\		$\vert$	
$n/4 \; n/4 \; n/4 \; n/4$	$4(n/4)$	$n/4$	$n/4$

$1 + \log_2(n)$

// Same amount of work @ each level

$\Sigma = (1 + \log_2(n))n + n + n\log_2(n)$

↳ $O(n \log_2 n)$

// Total work @ each level decreasing

$\Sigma = n + n/2 + n/4 + \ldots + 1$

$\Sigma = n(1 + \frac{1}{2} + \frac{1}{4} + \ldots (1/2)^{\log_2 n})$ //factor out n

↑ term that dominates

↳ $O(n)$

// Dividing by same amount of work, only thing diff : the constant

(3)

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

n — // work @ each level: $2^i n = 4^i (n/2^i)$

$n/2 \quad n/2 \quad n/2 \quad n/2 \qquad 4(n/2) = 2n$

$n/4 \quad n/4 \ldots \ldots \qquad 16(n/4) = 4n$

Solution to recurrence:

$$T(n) = \sum_{i=0}^{\log_2 n} 2^i \times n \qquad \boxed{\sum_{i=0}^{n} 2^i = 2^{i+1} - 1} \text{ // given formula}$$

$$= (2^{\log_2(n)+1} - 1) \times n$$

$$= (2^{\log_2(n)} \times 2^1 - 1) n \quad // 2^{\log_2(n)} = n$$

$$= (2n - 1) n$$

$$= 2n^2 - n$$

$$\sim 2n^2$$

$$= O(n^2)$$

Three cases

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

(2) → if $a < 2$ $T(n) > \Theta n$ // decreasing

(1) → $a = 2$ $T(n) = \Theta(n \log(n))$ // same

(3) * $a > 2$ $T(n) = \Theta(n^{\log_2 a})$ // increasing

} amounts of work

↓

why?

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

$\underbrace{o \ldots o}_{a}$ // each node $n/2$

$\underbrace{\ldots \ldots}_{a^i}$ // each node is $n/2^i$

} $1 + \log_2(n)$

Why? cont

$$\sum_{0}^{\log_2(n)} n\left(\frac{a}{2}\right)^i$$

$$= n \sum_{0}^{\log_2 n} \left(\frac{a}{2}\right)^i$$

$$= \frac{n\left(\frac{a}{2}\right)^{\log_2(n)}}{2^{\log_2 n}}$$

$$= a^{\log_2(n)}$$

$$\left(\frac{a}{2}\right)^0 < \left(\frac{a}{2}\right)^{\log_2(n)}$$

$a = 2^{\log_2 a}$

$(2^{\log_2 a})^{\log_2(n)}$

$2^{\log_2 a \times \log_2(n)}$

$2^{\log_2 n \times \log_2 a}$

$n^{\log_2 a}$

general case for any $n > n$

Master Theorem (MT)

$$T(n) \begin{cases} aT(n/b) + n^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$$

cases:

1) $\log_b a < c$ $\quad T(n) = O(n^2)$
2) $\log_b a = c$ $\quad T(n) = O(n^2 \log_b(n))$
3) $\log_b a > c$ $\quad T(n) = O(n^{\log_b n})$

Applying Case 3 of MT:

Given: $T(n) = 8T(n/2) + 1000n^2$

$\log_2 8 = 3$ // $\log_b a$

$\underbrace{3}_{a} > \underbrace{2}_{c}$ $\quad a > c$

$T(n) = O(n^{\log_2 8}) = O(n^3)$

$a = 8$

$b = 2$

$c = 2$

$f(n) = 1000n^2$

$f(n) = O(n^2)$

Applying Case 2 of MT

1

$T(n) = 2T(n/2) + 10n$

$\log_2 2 = 1$ // $\log_b a = c = 1$

$T(n) = O(n^2 \log_2 n) = O(n \log_2 n)$

$a = 2$

$b = 2$

$c = 1$

?

2 $T(n) = n + 10n\log_2(n)$

$= O(n \log_2 n)$

Applying Case 1 of MT

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

$\log_2 2 = 1$

$1 < c = 2$

$T(n) = O(n^c)$

$= O(n^2)$

$a = 2$

$b = 2$

$c = 2$

$f(n) = n^2$

$f(n) = O(n^2)$

$$\boxed{\begin{aligned} T(n) &= 2n^2 - n \\ &\sim 2n^2 \\ &= n^2 \end{aligned}}$$

Quiz & Practice ~ mergesort

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

$a=2 \quad b=2 \quad c=1$

$\log_2 2 = 1$

$1 = 1$

$T(n) = O(n^2 \log_2(n))$

$= O(n \log_2 n)$

Karatsuba ~ multiplying 2 large ints

$$\begin{array}{r} 4102 \\ \times \; 1819 \\ \hline 3918 \\ 4102 \\ 32816 \\ 4102 \end{array}$$

n = # of digits

n levels

$\boxed{41}\ \boxed{02} \rightarrow 10^2 \boxed{41} + \boxed{02}$ (a, b)

$\boxed{18}\ \boxed{19} \rightarrow 10^2 \boxed{18} + \boxed{19}$ (c, d)

$(10^{n/2} a + b) \times (10^{n/2} c + d)$

$= 10^n ac + 10^{n/2} ad + 10^{n/2} bc + bd$

$= 10^n ac + 10^{n/2}(ad + bc) + bd$

How to represent as recurrence?

① Break # in 4 $n/2$

② Conquer

if $n > 1$, recursively ac, ad, bc, bd

③ If $n = 1$

$10^n ac + 10^{n/2}(ad + bc) + bd$

linear time shift

$T(n) = 4T(n/2) + 5n$ | $a = 4$

$\log_2 4 = 2$ | $b = 2$

$2 > 1$ | $c = 1$

$T(n) = O(n^{\log_2 4})$

$= O(n^2)$

$\boxed{10^n ac + 10^{n/2} \underbrace{(ad + bc)} + bd}$

if we can write the terms as ac or bd.

we don't need to do complex × //reduces recursive step

trick : $ad + bc = (a+b)(c+d) - ac - bd$

how? $ac + ad + bc + bd - ac - bd$

$T(n) = 3T(n/2) + 8n$

$\log_2 3 > 1$

$O(n^{\log_2 3})$

Ex: $aT(n/2) + n^c$ Proof for case 1

$\log_b(n)$

a^i sub problem

$(a/b^i)^c$

$= a^i(n/b^i)^c$

$= n^c(a/b)^i$

$\sum_{i}^{\log_b n} n^c(a/b)^i$

$n^c \sum_{i=0}^{\log_b n} (a/b)^i$

$= O(n^c)$