1002254987

Neel Katrodiya

Ch-4 HW

4.3-1 Show that the solution of
$T(n) = T(n-1) + n$ is $O(n^2)$

→ Here, We have to prove that $T(n) = T(n-1) + n$
is $O(n^2)$.
We make a guess.
$T(n) \leq cn^2$
where $c > 0$

$T(n-1) \leq c(n-1)^2$

Substituting in equation
Inductive Step
$T(n) \leq c(n-1)^2 + n$

$\therefore T(n) \leq c(n^2 - 2n + 1) + n$
$\therefore T(n) \leq cn^2 - 2nc + c + n$
$\therefore T(n) \leq cn^2 + (-2c + 1)n + c$
$\leq cn^2$

Specifically, we require

$-2c + 1 \leq 0$ or
$c \geq \frac{1}{2}$

So that $T(n) = T(n-1) + n$ is $O(n^2)$.

Q-4.4 ③ Given, $T(n) = 4T(n/2 + 2) + n$

$i=0$; $T(n) = 4T(n/2) + n$
∴ backward substitution

$i=1$; $T(n/2) = 4T(n/4) = n/2$

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

$$= 16T(n/4) + 2n + n$$

$i=2$; $T(n/4) = 4T(n/8) + n/4$

$$T(n) = 16[4T(n/8) + n/4] + 2n + n$$

$$= 64T(n/8) + 4n + 2n + n$$

∴ It can be written as,

$$T(n) = n \sum_{i=0}^{\log n - 1} 2^i + 4^{\log n} T(1)$$

$$T(n) = \Theta(n^2) + n\left[\sum_{i=0}^{\lg n - 1} 2^i\right]$$

$$= \frac{2^{\lg n} - 1}{2^{-1}}$$

$$= 2^{\lg n} - 1$$

$$= n - 1$$

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

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

Q-4.5 Use the master method to give tight asymptotic bounds for the following recurrences.

a) $T(n) = 2T(n/4) + 1$

Master's ~~Med~~ Method is

$$T(n) = aT(n/b) + \Theta(n^d)$$

$$a \geq 1, \quad b > 1, \quad d \geq 0$$

Then case 1: $T(n) = \Theta(n^d)$ if $a < b^d$

case 2: $T(n) = \Theta(n^d \lg n)$ if $a = b^d$

case 3: $T(n) = \Theta(n^{\log_b a})$ if $(a > b^d)$

a) $T(n) = 2T(n/4) + 1$

$a = 2,\ b = 4,\ d = 0 \quad (\because n^0 = 1)$

$\Rightarrow 2 > 4^0$

$\Rightarrow 2 > 1$

Then $T(n) = \Theta(n^{\log_4 2})$

So, solution is $T(n) = \Theta(\sqrt{n})$

b) $T(n) = 2T(n/4) + \sqrt{n}$ $\quad$ ak + $T(n) = 4T(n/0)$

① $a = 2,\ b = 4,\ d = 1/2 \quad (\because n^{1/2})$ $f(n)$

$\Rightarrow 2 = 4^{1/2} \quad \Rightarrow 2 = 2$

$\therefore T(n) = n^{\log_a b}$

Then $T(n) = \Theta(n^{\log_4 2} \log n)$

$T(n) = \Theta(\sqrt{n} \log n)$ → by master's ~~the~~ theorem

② $T(n) = 2T(n/4) + m$, here $m = \sqrt{n}$

$a = 2$, $b = 4$; $f(n) = m$

$T(n) = \sqrt{n} + m$ (non recursive)

$= M + M$ (recursive part are equal so we add $\log n$)

$= m + \log n$

we $k + M = \sqrt{n}$

$T(n) = \Theta(\sqrt{n} \log n)$

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

$a = 2$, $b = 4$, $d = 1$

$\therefore 2 < 4$

then $T(n)$ $\Theta(n^d)$

Solution is $T(n) = \Theta(n)$

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

$a=2,\ b=4,\ d=2$

$2 < 4^2$

then $T(n) = \Theta(n^d)$

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

(1) Given the code to calculate $n!$

$T(n) = T(n-1) + C$

backward substitution,

$i=1;\ T(n) = T(n-1) + c$

$i=2;\ T(n) = T(n-2) + 2c$

$i=3;\ T(n) = T(n-3) + 3c$

We can see that,

$T(n) = T(n-3) + ic \qquad ,\ i = 1, 2, \ldots, n-1$

till $T(1) = \Theta(1)$

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

$\boxed{T(n) = \Theta(n)}$

(2) Given the code to calculate a fibonacci number.

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

Ex. $fib(0) = 0$

$fib(1) = 1$

$fib(2) = fib(1) + fib(0)$

$= 1 + 0$

$= 1$

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

$T(n) = 2T(n-2)$ at $i=1$

$T(n) = 4T(n-4)$ at $i=2$

$T(n) = 8T(n-6)$ at $i=3$

$T(n) = 2^iT(n-2i)$ (∵ We can see that)

$T(1) = n - 2i = 0$

$i = n/2$

$T(n) = 2^{n/2}\, T(n - 2(n/2))\, T(0)$

$\boxed{T(n) = \Omega(2^{n/2})}$