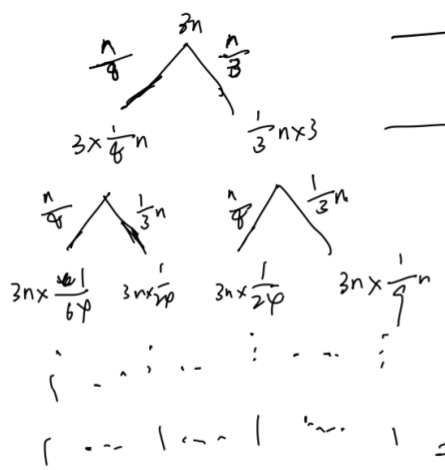
1. First use the iteration method to solve the recurrence, draw the recursion tree to analyze.

$$T(n) = T(\tfrac{n}{8}) + T(\tfrac{n}{3}) + 3n$$

Then use the substitution method to verify your solution.

Friday, February 11, 2022 11:21 PM

1. $T(n) = T(\frac{n}{8}) + T(\frac{n}{3}) + 3n$



— $3n$

— $\frac{11}{8}n = (\frac{1}{8}+\frac{1}{3})\cdot 3n$

→ $3n[(\frac{1}{8})^2+(\frac{1}{3}\times\frac{1}{8})+\frac{1}{3}\times\frac{1}{8}+(\frac{1}{3})^2]$

$= 3n\cdot(\frac{1}{8}+\frac{1}{3})^2$

$\Rightarrow$ $= 3n\cdot(\frac{1}{8}+\frac{1}{3})^k$

The time complexity is $T(n)$: $\Theta(n)$

with the subtitution method

$\Rightarrow$ upper bound

$T(k) \le dk$

$\Rightarrow T(\frac{n}{4}) + T(\frac{n}{3}) + 3n \le d \cdot \frac{n}{4} + d \cdot \frac{n}{3} + 3n$

$\Rightarrow$ if $d \ge \frac{72}{13}$, $T(n) \le dn \Rightarrow T(n) = \Theta(n)$

$\Rightarrow$ lower bound

$T(k) \ge ck$,

same procedure

$\Rightarrow$ if $c \le \frac{72}{13}$, $T(n) \ge cn$. $T(n) = \Omega(n)$

2. Use the substitution method to prove that $T(n) = 2T(\frac{n}{2}) + cnlog_2 n$ is $O(n(log_2 n)^2)$.

Question 2.

Base case $T(1) = 1 > 0 = d \times 1 \times \log_2 1$

$T(2) \leq d \times 2 (\log_2 2)^2$

Induction $\Rightarrow$ If for all $k < n$, we have $T(k) \leq dk(\log_2 k)^2$

$\Rightarrow T(n) = 2T(\frac{n}{2}) + cn\log_2 n \leq 2d\frac{n}{2}(\log_2 \frac{n}{2})^2 + cn\log_2 n$

$= dn(\log_2 n)^2 - 2dn\log_2 n + dn + cn\log_2 n$

$\Downarrow$

~~If~~ ...... ≤ 0

$$\Rightarrow \quad d \geq \frac{c \log_2 n}{2 - \frac{1}{\log_2 n}} \leq c \frac{1}{2 - \frac{1}{\log_2 2}} = c$$

$$\Rightarrow \quad d \geq c. \quad T(n) \leq dn(\log_2 n)^2$$

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

3. Solve the recurrence:

   $T(n) = 3T(\sqrt{n}) + (\log n)^2$

   (Hint: Making change of variable)

Question 3.

$$T(n) = 3T(\sqrt{n}) + (\log_2 n)^2$$

set $k = \log_2 n$

$$2^k = n$$

$\therefore$ we have $\cancel{T(n)}$ $T(2^k) = 3T(2^{\frac{k}{2}}) + k^2$

$\Rightarrow$ set $y(k) = T(2^k)$

$$= 3f(\tfrac{k}{2}) + k^2$$

$$= \Theta(k^{\log_2 3} \cdot \log k) + \Theta(k^2)$$

$\Rightarrow$ $y(k) = \Theta(k^2)$

$\because$ $k = \log_2 n$

$\Rightarrow$ $T(n) = T(2^k) = \Theta(k^2) = \Theta(\log_2 n)^2$

4. You have three algorithms to a problem and you do not know their efficiency, but fortunately, you find the recurrence formulas for each solution, which are shown as follows:

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

B: $T(n) = 2T(\frac{9n}{10}) + \theta(n)$

C: $T(n) = 2T(\frac{n}{2}) + \theta(n^2)$

Please give the running time of each algorithm (In θ notation), and which of your algorithms is the fastest (You probably can do this without a calculator)?

Question 4

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

first we set $a=2$, $b=2$, $d=1$

$$\frac{a}{b^d} = \frac{2}{2^1} = 1$$

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

B:

for $T(n) = 2T(\frac{9n}{10}) + \Theta(n)$

$\Rightarrow$ we set $a = 2$, $b = \frac{10}{9}$, $d = 1$

Base on the formula $\frac{a}{b^d} = \frac{2}{\frac{10}{9}} = \frac{9}{10} \times 2$

$= \frac{18}{10}$

$= \frac{9}{5} > 1$

$\therefore$ $T(n) = \Theta(n^{\log_b(a)}) = \Theta(n^{\log_{\frac{10}{9}}(2)})$

C.

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

we set $a=2$, $b=2$, $d=2$

$$\Rightarrow \frac{a}{b^d} = \frac{2}{2^2} = \frac{2}{4} = \frac{1}{2} < 1$$

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

The A algorithm is the fastest