Recurrence Relation

Substitution relation

1) $T(n) = \begin{cases} n * T(n-1) & \text{if } n=1 \\ 1 & \text{if } n \geq 1 \end{cases}$

$T(n) = n * T(n-1)$ — (a)

$T(n-1) = (n * T(n-1)) - 1$

$= (n-1) * T(n-2)$ — 1

$T(n-2) = (n-2) * T(n-3)$ — 2

putting in (a)

$T(n) = n * (n-1) * (n-2) * T(n-3)$

$= n * (n-1) * (n-2) * \ldots * T(1)$

$= n * (n-1) * (n-2) * \ldots 1$

$= n * n(1 - \frac{1}{n}) * n(1 - \frac{2}{n}) * \ldots n(\frac{1}{n})$

$n * n * n = n^3$

$= O(n^n)$

$T(n) = \begin{cases} T(n-1) + 1 \\ 1 \text{ if } n = 0 \end{cases}$

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

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

$T(n-2) = T(n-3) + 1$

putting in orignal eq

$= T(n-2) + 2$

$= T(n-3) + 3$

$= T(n-k) + k$

let $n-k=0$

$\therefore n=k$

$= T(n-n)+n$

$= T(0)+n$

$= 1+n$

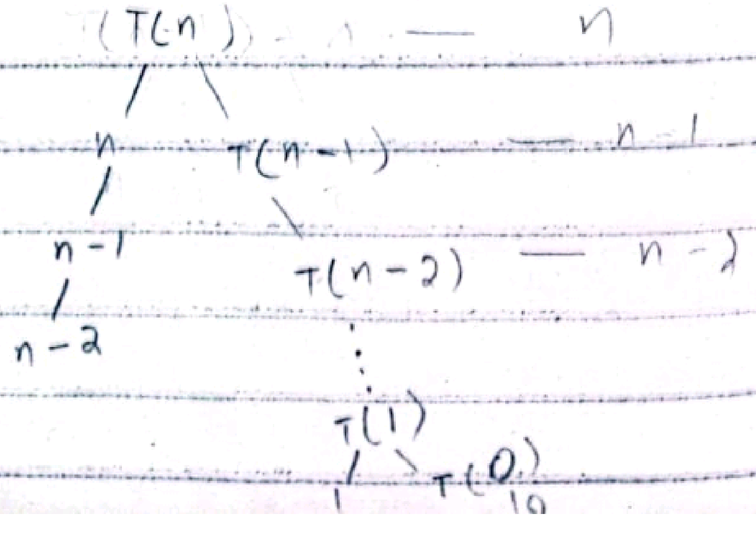
$T(n) = O(n)$

3

```
void T(int n) {
  if (n>0) {                       — 1
     for(i=0; i<n; i++) {          — n+1
         print()  }                — n
     T(n-1)                        — T(n-1)
  }
}
```

$T(n) = T(n-1) + 2n + 2$
constant

$$T(n) = \begin{cases} 1 & n=0 \\ T(n-1)+n & \end{cases}$$



$n + (n-1) + (n-2) + (n-3) + \dots 1$

$0 + 1 + 2 + 3 + \dots + n - 1$

$n(n-1)/_2 = O(n^2)$

(4) void Test(int n){

if(n>0){

for(i=1; i<n; n*2){

print(); } $-\log n$

Test(n-1) $-$ $T(n-1)$

}

$T(n) = \begin{cases} T(n-1) + \log n & n>0 \\ 1 & n=0 \end{cases}$

$= T(n-1) + \log n$

$T(n-1) = T(n-2) + \log n \quad -(1)$

$T(n-2) = T(n-3) + \log n \quad -(2)$

$= T(n-2) + 2\log n$

$= T(n-3) + 3\log n$

$T(n-k) + k\log n$

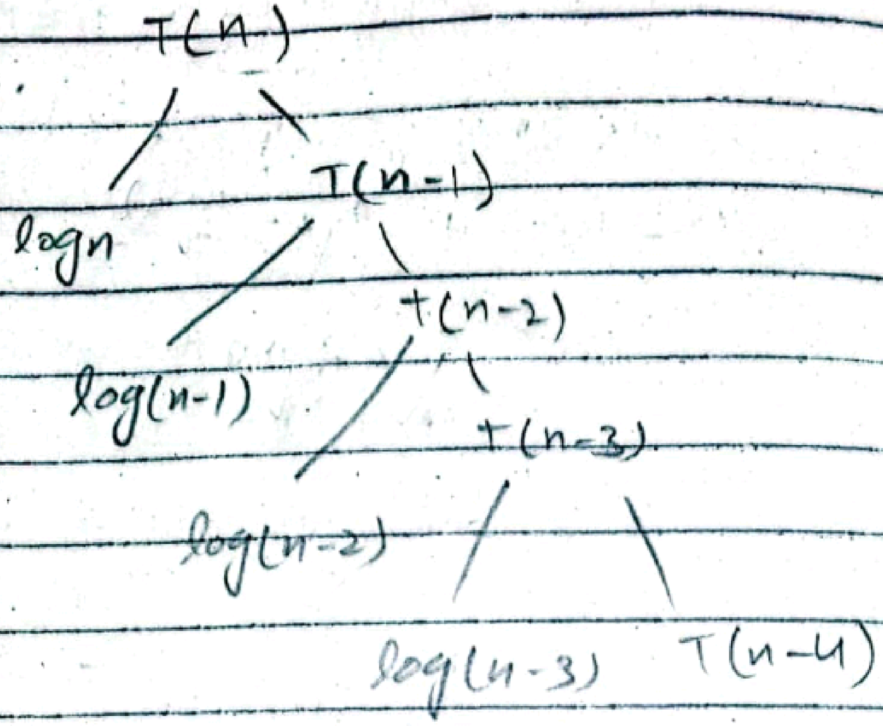
$n-k=0$

$n=k$

$= T(0) + n\log n = 1 + n\log n$

$= O(n \log n)$

2) Using Tree method



$= \log n + \log(n-1) + \log(n-2) + \cdots \log 0$

$= \log [n + (n-1) + (n-2) + (n-3) + \cdots 1]$

$= \log [n + n-1 + n-2 + n-3 + \cdots 1]$

$= \log n!$

$= \log n^n$

$= n \log n$

$= O(n \log n)$

Recurrence Observation

$T(n) = T(n-1) + 1$ — $O(n)$

$T(n) = T(n-1) + n$ — $O(n^2)$

$T(n) = T(n-1) + \log n$ — $O(n \log n)$

n is multiplying by last, no co-efficient on left side.

$2T(n-1) + 1$

case is different

n becomes power of co-efficient

$2T(n-1) + 1$ — $O(2^n \times 1)$

$3T(n-1) + 1$ — $O(3^n)$

$2T(n-1) + n$ — $O(n 2^n)$

Master theorum for decreasing function

$T(n) = aT(n-b) + f(n)$

$a > 0$, $b > 0$ & $f(n) = O(n^k)$

where $k \geq 0$

1) if $a = 1$ $O(n^{k+1}) = O(n * f(n))$

2) if $a > 1$ $O(n^k a^{n/b})$

Example

$2T(n-2) + 1$ — $O(2^{n/2})$

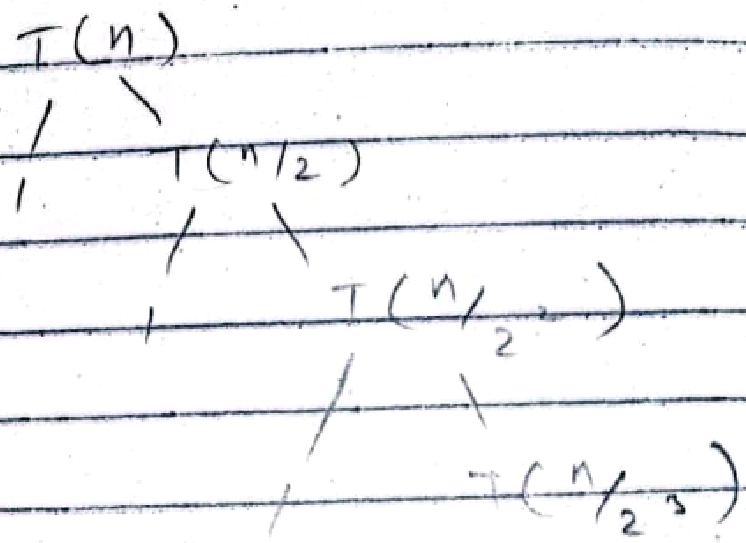
$a = 2$, $b = 2$, $f(n) = 1$

3) if $a < 1$ $\quad O(n^k) = O(f(n))$

$-2T(n-1) + 2 \quad O(n)$

$f(n) = n^0 =$

5) $T(n) = \begin{cases} 1 & n = 1 \\ T(n/2) + 1 & n \geq 1 \end{cases}$



$T(n/2^k)$

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log_2 n$

2.) using Substitution

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

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

$T(n/4) = T(n/8) + 1$

putting values:

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

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

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

let $\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log_2 n$

$$= T(1) + \log n$$

$$= 1 + \log n$$

prove that code is correct:

$\{x = v_x, y = v_y\}$

temp = x;

x = y;

y = temp;

$\{x = v_y, y = v_x\}$

Solution:

$\{x = v_x, y = v_y\}$ temp = x; $\{x = v_x, y = v_y$

$\{x = v_x, y = v_y\}$ x = y; $\{x = v_y, y = v$

(1) Substitution method
version #102

$n/4$ $\times 6?$ $8 \times \frac{4}{1}^2$ $\frac{n/2}{\frac{2}{1}}$ $\frac{n}{2} \times \frac{1}{2}$

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

$$T(n/2) \quad \cancel{=} 8\left[8T\left(\frac{n}{4}\right) + \frac{n^2}{2}\right] + n^2$$

$$T(n/2) = 8T(n/4) + \left(\frac{n}{2}\right)^2$$

$$T(n/8) = 8T(n/2\cancel{4}) + (n/8)^2$$

$$T(n/4) = 8T(n/2^3) + \left(\frac{n}{2^2}\right)^2$$

Substituting $T(n/2)$

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

$$= 8\left[8T(n/4) + \left(\frac{n}{2}\right)^2\right] + n^2$$

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

Substituting $T(n/4)$

$$T(n) = 8^2\left[8T(n/2^3) + (n/4)^2\right] + 2n^2 + n^2$$

$$= 8^3 T(n/2^3) + 4n^2 + 2n^2 + n^2$$

$$\Rightarrow 8^3\left[8T(n/2^4) + \left(\frac{n}{2^3}\right)^2\right] + 4n^2 + 2n^2 + n^2$$

$$\Rightarrow 8^4 T(n/2^4) + 8n^2 + 4n^2 + 2n^2 + n^2$$

$$\Rightarrow 8^4 T(n/2^4) + 2^3 n^2 + 2^2 n^2 + 2^1 n^2 + 2^0 n^2$$

$$\Rightarrow 8^k T(n/2^k) + 2^{k-1} n^2 + 2^{k-2} n^2 + \cdots 2^1 n^2 + 2^0 n^2$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$k = \log_2 n \text{ or } \log n$$

$$8^k = (2^k)^3$$

$$(2^k)^3 \times T(1) + n^2 [2^0 + 2^1 + 2^2 + \cdots 2^{k-1}]$$

$$(2^k)^3 * n^2 [\underbrace{2^0 + 2^1 + 2^2 + \cdots 2^{k-1}}_{\text{G.P series}}]$$

$$S = \frac{a(r^n - 1)}{r-1}$$

$$r = \frac{a_2}{a_1} = \frac{2^1}{2^0} = 2$$

$$a = 2^0 = 1$$

$$n = k$$

$$n^3 + n^2 \left[\frac{1(2^k - 1)}{2-1} \right]$$

$$= n^3 + n^2 (2^k - 1)$$

$$2^k = n$$

$$= n^3 + n^2 (n-1)$$

$$= n^3 + n^3 - n^2$$

$$= 2n^3 - n^2$$

(2)

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

$$T(n/2) = 2T(n/4) + n/2^1 \quad \text{--- (1)}$$

$$T(n/2^2) = 2T(n/2^3) + n/2^2 \quad \text{--- (2)}$$

putting both equations

$$\Rightarrow 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2^1}\right] + n$$

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

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

~~Again~~ Applying eq (2)

$$= 2^2\left[2T(n/2^3) + n/2^2\right] + n + n$$

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

Applying other eq

$$= 2^3\left[2T\left(\frac{n}{2^4}\right) + \frac{n}{2^3}\right] + n + n + n$$

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

$4 = k$

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

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log n$

$$= 2^k T(1) + kn$$

$2^k = n$

$= n + n\log n$

$\Rightarrow O(n\log n)$

(3) $T(n) = 2T(n/2) + n\log n$

$$T(n/2^1) = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\log\frac{n}{2} \quad -(1)$$

$$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\log\frac{n}{2^2} \quad -(2)$$

$$T\left(\frac{n}{2^3}\right) = 2T\left(\frac{n}{2^4}\right) + \frac{n}{2^3}\log\frac{n}{2^3} \quad -(3)$$

putting values:

$$= 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\log\frac{n}{2}\right] + n\log n$$

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

putting eq (2)

$$= 2^2\left[2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\log\frac{n}{2^2}\right] + n\log\frac{n}{2} + n\log n$$

$$= 2^3 T\left(\frac{n}{2^3}\right) + n\log\frac{n}{2^2} + n\log\frac{n}{2} + n\log n$$

$$= 2^3\left[2T\left(\frac{n}{2^4}\right) + \frac{n}{2^3}\log\frac{n}{2^3}\right] + n\log\frac{n}{2^2} + n\log\frac{n}{2} + n\log n$$

$$= 2^4 T\left(\frac{n}{2^4}\right) + n\log\frac{n}{2^3} + n\log\frac{n}{2^2} + n\log\frac{n}{2} + n\log n$$

$$2^k T\left(\frac{n}{2^k}\right) + n\log\frac{n}{2^{k-1}} + n\log\frac{n}{2^{k-2}} + n\log\frac{n}{2} + n\log n$$

$$2^k \times T(1) + n\left[\log\frac{n}{2^{k-1}} + \log\frac{n}{2^{k-2}} + \dots \log\frac{n}{2^1} + \log\frac{n}{2^0}\right]$$

$$\log \frac{n}{2^{k-1}} = \log \frac{n}{2^k} \times \log \frac{\frac{1}{2^{-m}}}{2^k} = 1$$
$$= \log 2$$

$$\Rightarrow n + n\left[\log 2 + \log 4 + \dots \log \frac{n}{2} + \log n\right]$$

$$= n + n\left[1 + 2 + 3 + \dots + \log(n-1)\right]$$

$1 + 2 + 3 + \dots$

$$\frac{n(n+1)}{2}$$

$= \log \frac{n}{2}$

$= \log n - \log 2$

$1 + 2 + 3 + \dots \log(n-1) + \log n$

$$\frac{\log n(\log n + 1)}{2}$$

$$= n + n\left[\frac{\log n(\log n + 1)}{2}\right]$$

$$n + n\left(\frac{(\log n)^2 + \log n}{2}\right)$$

$$= n + \frac{n(\log n)^2 + n\log n}{2}$$

$$= O(n(\log n)^2) = O(n\log^2 n)$$

3)

$$T(n) = \begin{cases} 1 & n == 1 \\ 2T(n/2) + \frac{n}{\log n} & n > 1 \end{cases}$$

$T(n) = 2T(n/2) + n/\log n$ — (a)

$T(n/2) = 2T(n/2^2) + \frac{n/2}{\log n/2}$ — (1)

$T(n/2^2) = 2T(\frac{n}{2^3}) + \frac{n/2^2}{\log n/2^2}$ — (2)

$T(n/2^3) = 2T(n/2^4) + \frac{n/2^3}{\log n/2^3}$ — (3)

substituting values in (a)

$$T(n) = 2\left[2T(n/2^2) + \frac{n/2}{\log n/2}\right] + n/\log n$$

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

putting eq (2)

$$= 2^2\left[2T(n/2^3) + \frac{n/2^2}{\log n/2^2}\right] + n/\log n/2 + n/\log n$$

$$= 2^3 T(n/2^3) + \frac{n}{\log n/2^2} + n/\log n/2^1 + n/\log n$$

putting eq (3)

$$\Rightarrow 2^3\left[2T(n/2^4) + \frac{n/2^3}{\log n/2^3}\right] + n/\log n/2^2 + n/\log n/2 + n/\log n$$

$\Rightarrow 2^4 T(n/2^4) + n/\log n/2^3 + n/\log n/2^2 + n/\log n/2^1$
$+ n/\log 2^0$

$k = 4$

$\Rightarrow 2^k T(n/2^k) + n\left[1/\log 2^{k-1} + 1/\log 2^{k-2} + 1/\log^{k-3} + \dots + 1/\log \frac{n}{2} + 1/\log \frac{n}{2^0} \right]$

$n/2^k = 1 \quad 2^k = n$

$\Rightarrow \quad k = \log n$

$$n + n\left[\frac{1}{\log \frac{n}{2^{k-1}}} + \frac{1}{\log \frac{n}{2^{k-2}}} + \dots \frac{1}{\log \frac{n}{2^1}} + \frac{1}{\log \frac{n}{2^0}} \right]$$

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

$$= n + n\left[\frac{1}{\log_2 2} + \frac{1}{\log_2 4} + \frac{1}{\log 8} + \dots + \frac{1}{\log n/2^0} \right]$$

$$= n + n\left[\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \frac{1}{\log n/2^0} \right]$$

$$= n + n\left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{\log n - 1} + \frac{1}{\log n} \right]$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \frac{1}{n} \Rightarrow \log n$

$= n + n \log\log n$ $\qquad = \log \frac{n}{2^0}$

$= O(n \log\log n)$ $\qquad = \log n - \log 2^0$

4) $2T(n^{1/2}) + C$

$$T(n) = \begin{cases} 2 & 0 < n \le 2 \\ 2T(\sqrt{n}) + C & n > 2 \end{cases}$$

$T(n) = 2T(\sqrt{n}) + C$

$T(\sqrt{n}) = 2T(n^{\frac{1}{2^2}}) + C \quad —(1)$

$n^{1/2}$

$T(n^{1/2^2}) = 2T(n^{1/2^3}) + C \quad —(2)$

$T(n^{1/2^3}) = 2T(n^{1/2^4}) + C \quad —(3)$

$T(n) = 2T(n^{1/2}) + C$

$= 2\left[2\,T(n^{1/2^2}) + C\right] + C$

$= 2^2\,T(n^{1/2^2}) + 2C + 1C$

putting equation (2)

$= 2^2\left[2T(n^{1/2^3}) + C\right] + 2C + 1C$

$= 2^3\,T(n^{1/2^3}) + 2^2C + 2^1C + 2^0C$

eq (3)

$= 2^3\left[2T(n^{1/2^4}) + C\right] + 2^2C + 2^1C + 2^0C$

$\Rightarrow 2^4\,T(n^{1/2^4}) + 2^3C + 2^2C + 2^1C + 2^0C$

$k = 4$

$\Rightarrow 2^k T(n^{1/2^k}) + 2^{k-1}c + 2^{k-2}c + \dots + 2^1 c + 2^0 c$

$n^{1/2^k} = 2$

$\log n^{1/2^k} = \log 2$

$\frac{1}{2^k} \log n = \log 2$

$\frac{1}{2^k} \log n = 1$

$\log n = 2^k$

$\log \log n = \log 2^k$

$\log \log n = k \log 2$

$\boxed{k = \log \log n}$

$2^k \times 2 + c\left[2^0 + 2^1 + 2^2 + \dots + 2^{k-1}\right]$

$a = 2^0 = 1$ $\quad = \frac{a(r^n - 1)}{r - 1} \quad = \frac{1(2^k - 1)}{2 - 1}$

$r = \frac{2}{1} = 2$

$n = k$ $\quad = 2^k - 1$

$2^k \times 2 + c(2^k - 1)$

neglect 'c'

$= 2^k \times 2 + \frac{2^k - 1}{2^k} = 3 \cdot 2^k = 3 \log n$

$= O(\log n)$

5) $T(n) = 2T(n^{1/2}) + \log n$

$$T(n) = \begin{cases} 2 & 0 < n \leq 2 \\ 2T(\sqrt{n}) + \log n & n > 2 \end{cases}$$

$T(n) = 2T(n^{1/2}) + \log n$

$T(n^{1/2^1}) = 2T(n^{1/2^2}) + \log n^{1/2}$ — (1)

$T(n^{1/2^2}) = 2T(n^{1/2^3}) + \log n^{1/2^2}$ — (2)

$T(n^{1/2^3}) = 2T(n^{1/2^4}) + \log n^{1/2^3}$ — (3)

putting eq (1)

$\Rightarrow 2\left[2T(n^{1/2^2}) + \log n^{1/2}\right] + \log n$

$\Rightarrow 2^2 T(n^{1/2^2}) + \log n + \log n$

putting eq (2)

$\Rightarrow 2^2\left[2T(n^{1/2^3}) + \log n^{1/2^2}\right] + \log n + \log n$

$\Rightarrow 2^3 T(n^{1/2^3}) + \log n + \log n + \log n$

$\Rightarrow 2^3\left[2T(n^{1/2^4}) + \log n^{1/2^3}\right] + \log n + \log n + \log n$

$\Rightarrow 2^4 T(n^{1/2^4}) + \log n + \log n + \log n + \log n$

$\Rightarrow 2^k T(n^{1/2^k}) + k\log n$

$n^{\frac{1}{2^k}} = 2$

$\log n^{1/2^k} = \log 2$

$\frac{1}{2^k} \log n = 1$

$\log n = 2^k$

$\log\log n = \log 2^k$

$= k \log 2$

$k = \log\log n$

$= 2^k \times 2 + \log\log n \times \log n$

$= O(\log\log n \times \log n)$

6) $\sqrt{2}\, T(n/2) + \sqrt{n}$

$$T(n) = \begin{cases} 1 & n=1 \\ \sqrt{2}\, T(n/2) + \sqrt{2} & n>1 \end{cases}$$

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

$T(n/2) = \sqrt{2}\, T(n/2^2) + \sqrt{n/2}$

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

$T(n/2^3) = \sqrt{2}\, T(n/2^4) + \sqrt{n/2^3}$

$$\Rightarrow (\sqrt{2})^2 \, T\left(n/2^2\right) + \sqrt{n} + \sqrt{n}$$

$$= (\sqrt{2})^2 \left[\sqrt{2} \, T\left(n/2^3\right) + \frac{\sqrt{n}}{\sqrt{2}^2} \right] + \sqrt{n} + \sqrt{n}$$

$$= (\sqrt{2})^3 \, T\left(n/2^3\right) + \sqrt{n} + \sqrt{n} + \sqrt{n}$$

$$= (\sqrt{2})^3 \left[\sqrt{2} \, T\left(n/2^4\right) + \frac{\sqrt{n}}{\sqrt{2}^3} \right] + \sqrt{n} + \sqrt{n} + \sqrt{n}$$

$$= (\sqrt{2})^4 \, T\left(n/2^4\right) + \sqrt{n} + \sqrt{n} + \sqrt{n} + \sqrt{n}$$

$$= (\sqrt{2})^k \, T\left(n/2^k\right) + k\sqrt{n}$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\log n = k$$

$$(\sqrt{2})^k \times 1 + \log n \sqrt{n}$$

$$= O(\sqrt{n} \log n)$$

Substitution method using induction:

$T(n) = 3T(\frac{n}{4}) + n^2$ Guess : $O(n^2)$

Hypothes : $T(k) \leq c \cdot n^2$ $k < n$

$k = \frac{n}{4}$

inductive step : $3T(\frac{n}{4}) + n^2 \leq 3 \cdot c n^2 + n^2$

$\leq 3cn^2 + n^2$

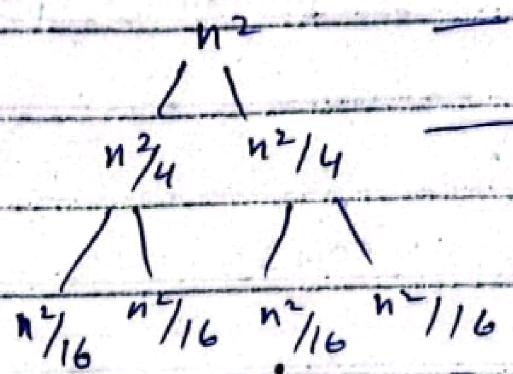
prove:

$T(1) = 1 \Rightarrow \begin{cases} T(2) = 3(\frac{2}{4}) + n^2 = 6 \\ T(3) = 3(\frac{3}{4}) + 3^2 = 11 \end{cases}$

$\left.\begin{aligned} 6 &\leq cn^2 = (6)^2 = 36 \\ 11 &\leq n^2 = (11)^2 = 121 \end{aligned}\right\}$ True

Recursive Tree method:

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



— n^2

— $n^2/2$

— $n^2/4$

$I_c = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \dots$

$n^2\left[1 + \frac{1}{2} + \frac{1}{4} + \dots \frac{1}{1-r}\right]$

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

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

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

$T(\frac{n}{2^k}) = T(1)$

$n = 2^k$

$\boxed{k = \log n}$

$I_c = kn^2$

$L_c = 2^k = n$

leaf node cost

$T(n) = L_c + I_c$

$= n + \cancel{k}n^2$

$= \cancel{n + n^2 \log}\ O(n^2)$

Recursion Tree method:

$T(n) = 3T(n/4) + n^2$

$T(n/4) = 3T(n/4^2) + (\frac{n}{4})^2$

$T(n/4^2) = 3T(n/4^3) + (\frac{n}{4^2})^2$

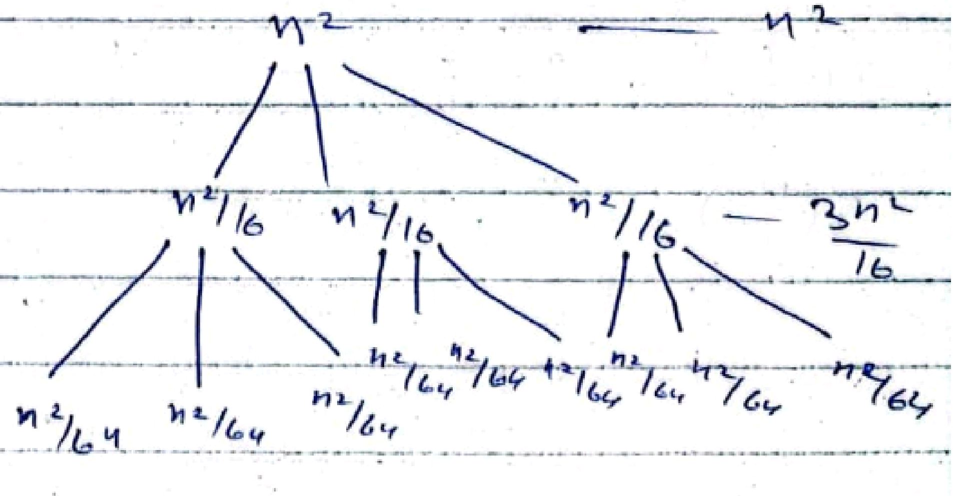
$\vdots$

$T(\frac{n}{4^k}) = T(1)$

$n = 4^k$

$\boxed{k = \log_4 n}$

Tree:



— $\frac{9n^2}{64}$

$T_c = \left[n^2 + \frac{3n^2}{16} + \frac{9n^2}{64} + \dots \frac{3^{k-1}n^2}{4^{k-1}}\right]$ — $\frac{3^k n^2}{4^k}$

$n^2\left[1 + \frac{3}{16} + \frac{9}{64} + \dots (\frac{3}{4})^{k-1}\right]$

$r < 1$

$\frac{1}{1-r} = \frac{1}{1-\frac{3}{16}} = \frac{1}{13/16} = \frac{16}{13}n^2$

$\boxed{T_c = \frac{16}{13}n^2}$

$L_c = 3^k = 3^{\log_4 n} \Rightarrow n^{\log_4 3}$

$T(n) = I_c + L_c \Rightarrow \frac{16}{13} n^2 + 3 n^{\log_4 3}$

$= \cancel{\Theta(3^n)}\ O(n^2)$

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

$T(\frac{n}{4}) = T(\frac{n}{16}) + T(\frac{n}{8}) + \frac{n^2}{16}$

$T(\frac{n}{2}) = T(\frac{n}{8}) + T(\frac{n}{4}) + \frac{n^2}{4}$

$T(\frac{n}{8}) = T(\frac{n}{32}) + T(\frac{n}{16}) + \frac{n^2}{64}$

$\cancel{T(\frac{n}{4})}$

$$n^2 \quad — \quad n^2$$

$$n^2/16 \qquad n^2/4 \quad — \quad \frac{5n^2}{16}$$

$$n^2/256 \quad n^2/64 \quad n^2/64 \quad n^2/16 \quad — \quad \frac{25}{256} n^2$$

$\vdots$

$T(1)$

$$\frac{5^{k-1}}{16^{k-1}} n^2$$

$T(\frac{n}{2^k}) = T(1)$

$k = \log n$

$L_c = 2^k = n$

$I_c =$

$$I_c = n^2 \left[\left(\frac{5}{16}\right)^0 + \left(\frac{5}{16}\right)^1 + \left(\frac{5}{16}\right)^2 + \cdots \left(\frac{5}{16}\right)^{k-1} \right]$$

$$= \frac{1}{1-r} = \frac{1}{1-\frac{5}{16}} = \frac{1}{11/16} = \frac{16}{11} n^2$$

$$\Rightarrow \frac{16}{11} n^2$$

$$T(n) = L_c + I_c = n + \frac{16}{11} n^2$$

$$\Rightarrow O(n^2)$$

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

(1) $T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$

(2) $T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$

(2) $T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$

(1) $T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$

(3) $T\left(\frac{n}{27}\right) = T\left(\frac{n}{81}\right) + T\left(\frac{2n}{81}\right) + \frac{n}{27}$

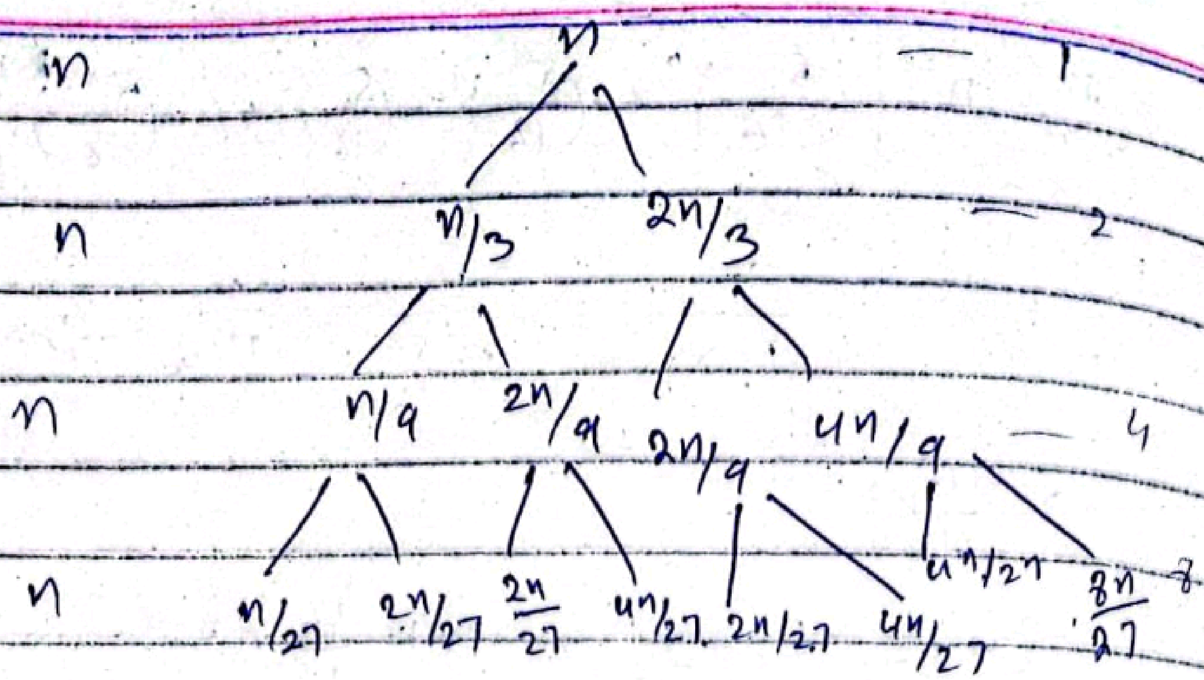
(3) $T\left(\frac{2n}{27}\right) = T\left(\frac{2n}{81}\right) + T\left(\frac{4n}{81}\right) + \frac{2n}{27}$

$\vdots$

$$T\left(\frac{2^k n}{3^k}\right) = T(1)$$

$$2^k n = 3^k, \quad n = \frac{3^k}{2^k}$$

$$k = \log_{3/2} n$$



$(k-1)n$

$$L_c = 2^k = 2^k = 2^{\log_{3/2} n}$$

$$= n^{\log_{3/2} 2}$$

$$I_c = kn = n \log_{3/2} n$$

$$T(n) = n \log_{3/2} n + n^{\log_{3/2} 2}$$

$\Rightarrow O(n \log n)$

Name: AOUN-HAIDER Section: A

ID: FA21-BSE-133

(Not registered)

Design & Analysis of Algorithms

Quiz: 02

(1)

Solve the recurrence using Master Theorem:

$$T(n) = 8T\left(\frac{n}{4}\right) + n^2$$

$4^2 = 1$

$a = 8$, $b = 4$, $f(n) = n^2$

$= n^{\log_b a} \Rightarrow n^{\log_4 8} \Rightarrow n^{1.-}$

Comparing $n^{\log_b a}$ & $f(n)$

$f(n)$ is greater, so case (3) ~~is~~ will be applied

1) Regularity verification:

$a f\left(\frac{n}{b}\right) < cf(n)$

$8 f\left(\frac{n}{4}\right) \leq cn^2$

$8 \times \frac{\cancel{n^2}}{16_2} \leq c\cancel{n^2}$

$\boxed{\frac{1}{2} \leq c}$

2) Complexity measurement:

According to case 3

$T(n) = O(f(n))$

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

Code snippet:

```
for (i=1; i<=n; i++){                    —— n
    for(j=1; j<=(n-i); j++)              —— Σ_{j=1}^{n-i} ti
        print("");                       —— Σ_{j=1}^{n-i} ti
    for (k=1; k<=(i-j); k++)             —— Σ_{k=1}^{i-j} ti
        print(" ");                      —— Σ_{k=1}^{i-j} ti
    print("");                           —— n
}
```

Num of steps

$$T(n) = n + 2\sum_{j=1}^{n-i} t_i + 2\sum_{k=1}^{i-j} t_i + n$$

$$= 2n + 2 \cdot \frac{(n-i)(n-i+1)}{2} + 2 \cdot \frac{(i-j)(i-j+1)}{2}$$

$$= \cancel{2n + n^2 - ni + n - ni + i^2 + i}$$

$$= 2n + n^2 - ni + n - ni + i^2 - \cancel{i} + i^2 - ij + \cancel{i} - ij + j^2 + j$$

$$= n^2 + 3n + 2i^2 + j^2 - 2ni - 2ij + j$$

$$T(n) = n^2 + 2i^2 + j^2 + 3n - 3ni - 2ij + j$$

$$\Rightarrow O(n^2)$$

Code #03

```
x = n;
while( x > 0) {                                   — n
    if ( x < i )                                  — n
                                                  (4n²+3n)
        sum += foo( x ) + foo ( x - 1 );          — n(2n+1+2n+1+1)

    else
        sum += bar(x);                            — 2n+1+1

    x = x - 1;                                    — n
}
```

$$\Rightarrow n + \frac{n^2+n}{2} + 1 \Rightarrow \frac{2n+n^2+n+2}{2} \Rightarrow \frac{n^2+3n+2}{2}$$

```
bar(a) {
    for( int i = 0;  i < n ; ++i )                — n

        for(int j = 0; j < i ; ++j )
            sum += a*(i+j);
    return sum;                                   — 1
}
```

$$\sum_{j=0}^{i} 1 = \frac{n(n+1)}{2}$$

```
foo(a) {        => 2n+1
        for( int i = 0; i < n; ++i )              — n
            sum += a*i;                           — n
        return sum;                               — 1
}
```

$$T(n) = n+n + \text{Max}(4n^2+3n, \ 2n+2) + n$$

$$= 3n + 4n^2 + 3n \Rightarrow 4n^2 + 6n$$

$$\Rightarrow O(n^2)$$

Akra Bazzi method

- Used to solve non-linear recurrances
- Divide & Conquer like binary search & merge sort not for $g(n) = 2n$

General Form:

$$T(n) = a_1 T(b_1 x + \varepsilon_1(x)) + a_2 T(b_2 x + \varepsilon_2(x))$$

$$+ \dots a_n T(b_n x + \varepsilon_n(x)) + g(x)$$

$$\sum_{i=1}^{n} a_i T(b_i x + \varepsilon_i(x)) + g(x)$$

$$T(x) = \theta\left(x^p + x^p \int_1^x \frac{g(u)}{u^{p+1}} du\right) \quad -(1)$$

$$p = a_1 b_1^p + a_2 b_2^p + \dots a_n b_n^p = 1$$

$$\sum_{i=1}^{n} a_i b_i = 1 \qquad p < g(n)$$

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

$a_1 = 1$
$b_1 = 1/2$
$g(n) = c$
$\varepsilon_1(x) = 0$

putting value in eq(1)
Finding value of 'p'

$$T(x) = 1\left(\frac{1}{2}\right)^p = 1$$

if $p = 0$

$T(x) = 1$

So $\boxed{p = 0}$

$$T(x) = \theta\left(x^{0} + x^{0} \int_{1}^{x} \frac{c}{u^{0+1}}\, du \right)$$

$$= \theta\left(1 + c\, |\log u|_{1}^{x} \right)$$

$$= \theta\left(1 + c\left[\log x - \log 1 \right] \right)$$

$$T(x) = \theta(\log x)$$

or

$$T(n) = \theta(\log n)$$

(2) $3T\left(\frac{n}{5}\right) + 2T\left(\frac{n}{5}\right) + \theta(n)$

$a_1 = 3, \quad a_2 = 2$

$b_1 = 1/5, \quad b_2 = 1/5$

$\varepsilon_1(n) = 0, \quad \varepsilon_2(n) = 0$

$g(n) = n$ or $g(u) = u$

$$T(n) = \theta\left(n^{p} + n^{p} \int_{1}^{n} \frac{g(u)}{u}\, du \right)$$

$p = ??$

$$= \sum_{i=1}^{n} a_i \cdot b_i^{p} = 1$$

$$= 3\left(\frac{1}{5}\right)^{p} + 2\left(\frac{1}{5}\right)^{p} = 1$$

$$= \frac{3+2}{5^p} = \frac{5}{5^p} = 5^{1-p}$$

if $p = 1$, $1 = 1$

$$T(n) = \theta\left(n^{1} + n^{1} \int_{1}^{n} \frac{u}{u^2}\, du \right)$$

$= \theta(n + n\int_1^n \frac{1}{u}\,du)$

$= \theta(n + n\,|\log u|_1^n)$

$= \theta(n + n[\log n - \log 1])$

$= \theta(n + n\log n)$

$\boxed{T(n) \Rightarrow \theta(n\log n)}$

(3) $2T(\frac{n}{4}) + 3T(\frac{n}{6}) + \theta(n\log n)$

$\sum_{i=1}^{n} a_i\,T(b_i n) + g(n)$

$a_1 = 2,\ a_2 = 3,\ b_1 = 1/4,\ b_2 = 1/6$

$g(n) = \theta(n\log n)$

$p = ?$ let $p = 1$

$2(\frac{1}{4}) + 3(\frac{1}{6}) = 1$

$\frac{24}{24} = 1$

so; $p = 1$

$= n^1 + n^1 \int_1^n \frac{g(u)}{u}\,du$

$= n + n \int \frac{\cancel{u}\log u}{u^2}\,du$

$= n + n \int \frac{\log u}{u}\,du$

$\longleftrightarrow$ integration by parts required

$$\int \frac{\log y}{y}\, dy \Rightarrow \int \frac{1}{y} \cdot \log y \, dy$$

$$u = \frac{1}{y}, \quad v = \log y$$

$$u = \cancel{-\frac{1}{y^2}\, dy}, \quad \int v\, dv = \log y$$

$$u\, dy = -\frac{1}{y^2}$$

$$\Rightarrow u \int v\, dx - \int\left(\int v\, dx \times u\, dx\right) dx$$

ILATE

- I — Inverse Trignometric
- L — logarithmic
- A — Algebric
- T — Trignometric
- E — exponential

OR

By substitution method

$$\int \frac{\log y}{y}\, dy$$

let $\log y = t$

$$dt = \frac{1}{y}\, dy$$

putting

$$\int t\, dt = \frac{t^2}{2} d + c \Rightarrow \frac{(\log y)^2}{2} = \frac{\log y^2}{2}$$

$$= \theta\left(n + n\left(\frac{\log 4^2}{2}\right)_1^n \right)$$

$$\theta\left(n + n\left(\frac{\log n^2}{2} - \frac{\log 1}{2}\right)\right)$$

$$= \theta(n \log n^2)$$

(3) $\frac{1}{4}T\left(\frac{n}{4}\right) + \frac{3}{4}T\left(\frac{3n}{4}\right) + 1$

$a_1 = \frac{1}{4}$, $a_2 = \frac{3}{4}$, $b_1 = \frac{1}{4}$, $b_2 = \frac{3}{4}$

$g(n) = 1$

$P = ?$

$$\sum_{i=1}^{k} a_i b_i^p = \frac{1}{4}\left(\frac{1}{4}\right)^p + \frac{3}{4}\left(\frac{3}{4}\right)^p =$$

let $p = 0$

$$\frac{1}{4} + \frac{3}{4} = 1$$

so, $P = 0$

$$= \theta\left(n^p + n^p \int_1^n \frac{g(n)}{n^{1+p}}\, dn\right)$$

$$= \theta\left(n^0 + n^0 \int_1^n \frac{1}{n}\, dn\right)$$

$$\theta\left(1 + \left|\log n\right|_1^n\right)$$

$$= \theta\left(1 + (\log n - \log 1)\right)$$

$$T(n) = \theta(\log n)$$