ALGORITHM ANALYSIS & DESIGN

Module 1 Part 3

CST306

Q1. Let f(n) = 7n + 4. Prove that this is of the order of O(n).

(3)

Sln $f(n) = 7n+4$

to prove $f(n) = O(n)$

defn: $\boxed{f(n) = O(g(n))}$ if there exist two const c, n_0 such that

$0 \leq f(n) \leq c \cdot g(n) \quad \forall\ n \geq n_0$

(1)

let $f(n) = 7n+4$; $g(n) = n$. — ①.

for $n \geq 1$

$$7n + 4 \leq 7n + 4n = 11n \Rightarrow C \cdot g(n)$$

$c = 11$

$n_0 = 1$

$g(n) = n$.

$\Rightarrow f(n) \leq 11n \quad \forall n \geq 1$

① $\leftarrow$ $\boxed{f(n) = O(n)}$ hence proved.

3 marks

Part B

Q2. Solve the following recurrence using recursion tree method:

a. $T(n)=T(n/2) + 1$, $T(1) = 1$

b. $T(n)=2T(n/2) + n^2$, $T(1) = 1$

(a). $T(n) = T(n/2) + 1$, $T(1) = 1$ → (1).

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

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

$T(n/4) = T(n/16) + 1$

$T(n) \longrightarrow 1 \longleftarrow n/2^0$

$T(n/2) \text{ — } 1 \longleftarrow n/2^1$

$T(n/4) \text{ — } 1 \longleftarrow n/2^2$

$\vdots$

$k \quad T(n/2^k) \qquad n/2^k$

$T(1) = 1$

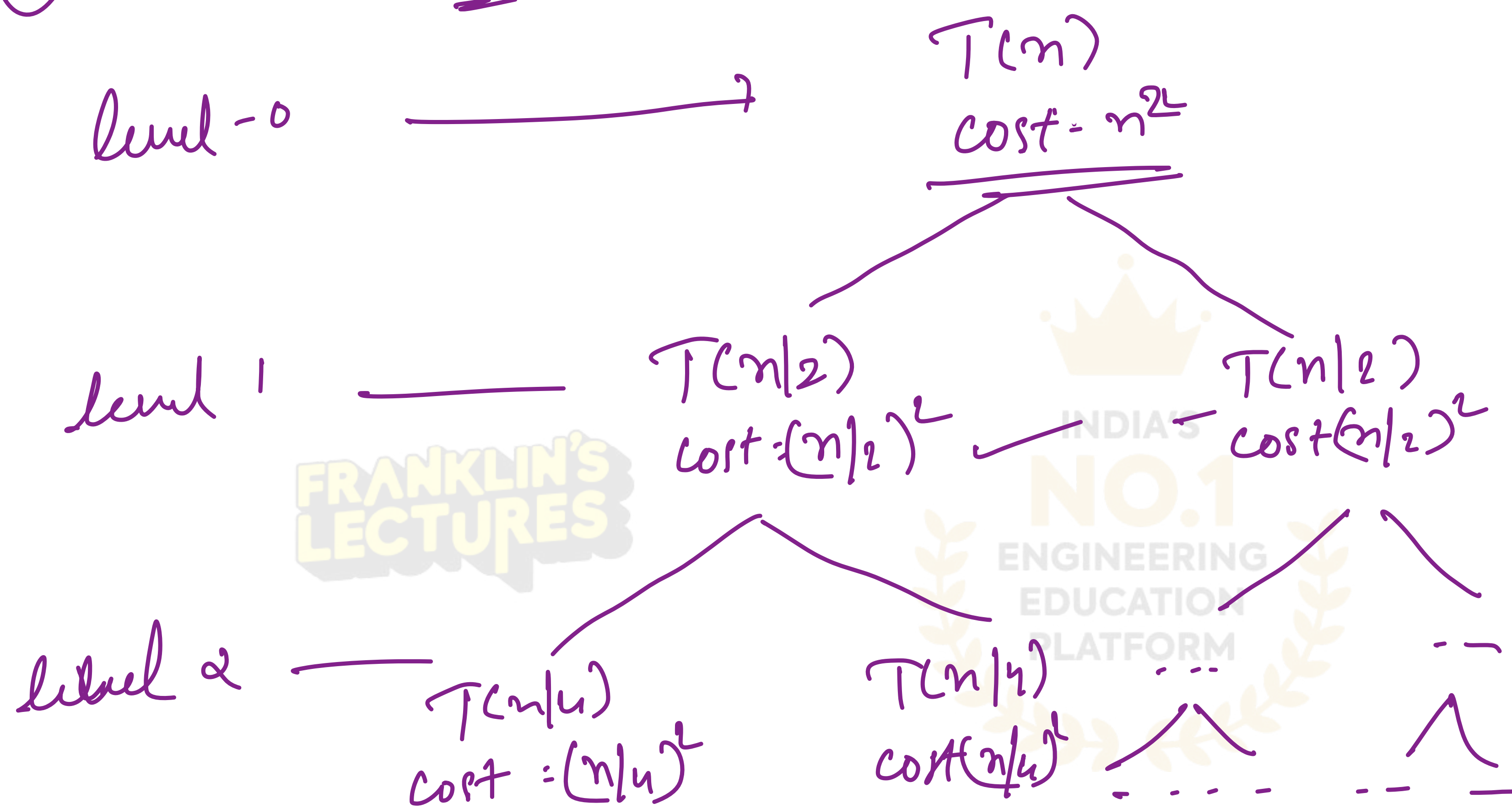
$n/2^k = 1$

$n = 2^k$

$\boxed{k = \log_2 n}$

$\boxed{T(n) = O(\log_2 n)}$

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



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

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

⋮

0 —— $n^2 \times 1 = n^2$ $\quad n^2/2^0$

1 —— $(n/2)^2 \times 2 = \dfrac{n^2}{2}$ $\quad n^2/2^1$

2 —— $(n/4)^2 \times 4 = n^2/4$ $\quad n^2/2^2$

$\vdots$

$\dfrac{n^2}{2^i} \Rightarrow n^2 + n^2/2 + n^2/4 + \dots$

$$\text{total cost} = n^2 \left[\underbrace{1 + \tfrac{1}{2} + \tfrac{1}{4} + \dots}_{GP} \right]$$

a – first term

r – common ratio

$a = 1$

$r = 1/2$

$$= n^2 \left[\frac{a}{1-r} \right]$$

$$= n^2 \left[\frac{1}{1 - 1/2} \right] = n^2 \left[\frac{1}{1/2} \right]$$

$$= n^2 \times 2$$

$= 2n^2 \quad c \cdot g(n)$

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

Q3. Solve the recurrence equation using Master's theorem.

a. T(n)=3T(n/2) + n2

b. T(n)=T(2n/3) + 1

Case 1 : if $a > b^k$ $T(n) = \Theta(n^{\log_b a})$ Imp

Case 2 : if $a = b^k$

a) $p > -1$ $T(n) = \Theta(n^{(\log_b a)} \cdot \log^{p+1}(n))$.

b) $p = -1$ $T(n) = \Theta(n^{\log_b a} \cdot \log(\log n))$

c) $p < -1$ $T(n) = \Theta(n^{\log_b a})$

Case 3 : if $a < b^k$

a) $p \geq 0$ $T(n) = \Theta(n^k \log^p(n))$

b) $p < 0$ $T(n) = \Theta(n^k)$

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

$\Rightarrow T(n) = aT(n/b) + f(n)$

$f(n) = \theta(n^k \log^p(n))$

$a = 3, \; b = 2, \; k = 2.$

$b^k = 4$

$a = b^k$ ✗ — case 2

$a > b^k$ ✗ — case 1

$\boxed{a < b^k \; \checkmark \text{ — case 3}}$

$P = 0 \quad P \geq 0$ ✓ case 3 (a)

Ans

$T(n) = \theta(n^k \log^p(n))$

$= \theta(n^2 \log^0(n))$

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

b. $T(n) = T(2n/3) + 1$

$\underline{n/b.}$

$a = 1$

$b = 3/2$

$f(n) = 1 \Leftarrow (n^{0} \log^{0} n)$

$k = 0$

$\underline{p = 0}$

$a = 1 \qquad b^k = (3/2)^0 = 1$

$a = b^k$ — case 2.

$0 > -1$

$p > -1 \qquad$ case 2 (a)

$T(n) = \Theta(n^{\log_b a} \log^{p+1}(n))$

$$\boxed{T(n) = \Theta(n^{\log_{3/2} 1} \log^{0+1}(n))}$$

Q5. Perform complexity analysis for the following code segments:

Nested loop

a. For i to n do

For j to n do A=B*C

End for

End for

for i to n do

for j to n do

A = B*C

end for

end for

outer loop – n times

inner loop – n times

A = B*C → 1 O(1)

⇒ n × n

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

Recursive f(n)

b. Function F(n)

{ If(n==0)

return(1)

Else

return (n*F(n-1))

}

function f(n)

{ if (n==0)

return (1).

} ← base condⁿ

else return n * f(n-1)

T(n-1)

n>0

f(0)

f(n) → f(n-1) → f(n-2) . . . f(0)

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

1) Complexity

2) Recursion
- Iteration | Substitution
- Recursion tree
- Master's theorem

3) Asymptotic notation

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

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

$= T(n/4) + 2$

$T(n/4) =$

$\vdots$

THANK YOU