Patient Demographics

Patient Name	Legal Sex	DOB	Address	Phone
Gandham, Sai	Male	7/1/1998	700 W MITCHELL CIR UNIT 1616 ARLINGTON TX 76013	682-377-9834 (Home) *Preferred*

Letter

THAM
TEXAS HEALTH ARLINGTON MEMORIAL TELEMETRY 3RD FLOOR
800 WEST RANDOL MILL ROAD
ARLINGTON TX 76012-2024
817-960-6100

Date September 6, 2022
Time 4:54 PM

To whom it may concern:

Sai Gandham was admitted to Texas Health Arlington Memorial through the emergency department on 09/05/2022 and has been under my care since 09/05/2022.

Thank you.

Bala Ponnam, MD
This note was electronically verified by the above THR Healthcare Provider.

HOME WORK

Name: Sai Rohit Kalyan Gandham.

StudentID: 1002070724

EmailID: sxg0724@mavs.uta.edu

2.21) Given to Express the function is

$n^3/1000 - 100n^2 - 100n + 3$

To prove: the Big O notation. (2)

* Ds per the time and Space Complexity the term which is having heighest power decides the the Big O(n)
* So given Expression is $n^3/1000 - 100n^2 - 100n + 3$

Case 1): Eq. take $n = 1$ then.

$(1^3/1000 - 100(1) - 100(1) + 3$

$= 1/1000 - 100 - 100 + 3$

$= 0.001 - 100 - 100 + 3$

$= 3.001 - 200 = -196.999$ → (1)

Case (2): if $n = 10$

$10^3/1000 - 100(10)^2 - 100(10) + 3$

$= 1 - 10000 - 1000 + 3$

$= 1 - 11000 = -10,998$

* from comparing (1) and (2) Equations we get to know the $O(n^3)$

3.1-1) Given $f(n)$ and $g(n)$ be asymptotically non-negative function.

To prove: $\max(f(n), g(n)) = \Theta(f(n)+g(n))$.

proof: also given non-negative function $\Rightarrow$

* We can say $f(n) \geq 0$ and $g(n) \geq 0$
  i.e $\Rightarrow$ if $f(n) \geq 0$ then $n \geq n_0$ ↳②
  ↳①
  n_0 - negative/

* So we can say By constants notation, Boundness

$$0 \leq c_1(f(n)+g(n)) \leq \max(f(n), g(n)) \leq c_2(f(n)+g(n))$$

* from Eq ① and ② we say.

$$f(n) \leq \max(f(n), g(n)) \rightarrow ④$$
$$g(n) \leq \max(f(n), g(n)) \rightarrow ⑤$$

* Adding Eq ④ + ⑤ $\Rightarrow$

$$f(n)+g(n) \leq 2\max(f(n), g(n))$$
$$\Rightarrow \quad 1/2(f(n)+g(n)) \leq \max(f(n), g(n))$$

$\Rightarrow$ Comparing with constant equations we can say

$$0 \leq 1/2(f(n)+g(n)) \leq \max(f(n), g(n)) \leq f(n)+g(n).$$

Hence proved that

$$\max(f(n), g(n)) = \Theta(f(n)+g(n)).$$

3.1-2)

Given to Show that $\boxed{(n+a)^b = \Theta(n)^b}$ → (1)

also given (a, b) are real numbers and $b > 0$.

Sol)

= let $a = 1$, $b = 1$ keep in Eqn (1)

Case 1 $(n+1)^1 = (n+1)$.

Case 2 $a = 2$, $b = 1$

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

Case 3 $a = 1$ & $b = 2$ $\Rightarrow$ $(n+1)^2 =$

$\Rightarrow$ $(n+1)^2 = n^2 + 2n + 1$

$= \Theta(n^2)$

* Note: We know that Big(O)(n) will be the height power of the Expression.

Case 4 $a = 1$, $b = 3$ $\Rightarrow$ $(n+1)^3$

$\Rightarrow n^3 + (1)^3 + 2(n)^2(1) + 2(n)(1)^2$

$\Rightarrow n^3 + 2n^2 + 2(n) + 1$

$\Rightarrow n^3 + 2n^2 + 2n + 1$

$\Theta(n^3)$

* from the case (1), (2), (3), and (4) We can able say that (a) is neglible so

... Hence proved $(n+a)^b \Rightarrow (n)^b$

$\Rightarrow \Theta(n)^b$

3.4 (b)

Sol) Given $f(n) + g(n) = \Theta(\min(f(n), g(n))$

let take $f(n) = n^3$

$g(n) = n^2$

So keeping $n^3 + n^2 = \min(n^3, n^2)$

$n^3 + n^2 \neq n^2$

if $n = 1$ $\quad (1)^3 + (1)^2 \neq (1)^2$

$\Rightarrow 2 \neq 1$ So false.

Hence $f(n) + g(n)$ is not Equal to $\Theta(\min(f(n), g(n))$

3.4 (H)

Given $f(n) + o(f(n)) = \Theta(f(n))$

let us assume $g(n) = o(f(n))$

$\Rightarrow 0 \leq g(n) \leq f(n) \quad n \geq n_0$

$\Rightarrow f(n) \leq f(n) + g(n) \leq f(n) + c f(n)$

$f(n) \leq f(n) + g(n) \leq (1 + c) f(n)$

Hence we can say that "True".

$f(n) + o(f(n)) = \Theta(f(n))$.

4. 4.2

Sol) Given to find the recursion tree upper bound on $T(n) = T(n/2) + n^2$

Need to Solve by using Substitution method.

Rate of increasing in No. of subproblem = 1
Rate of decreased in sub-problem size = 2.
depth of i: 0, 1, 2, 3 ... log n
Node cost = $(n/2^i)^2$
Hence the total cost of the tree is

$$T(n) = \sum_{i=0}^{\log n} (n/2^i)^2$$

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

$$\leq n^2 \sum_{i=0}^{\infty} (1/4)^i$$

$$= O(n^2)$$

Using Substitution Verifing

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

$\leq c(n/2)^2 + n^2$

$= cn^2/4 + n^2$

$= (c/4 + 1)n^2$

$\leq cn^2$

$= O(n^2)$

4.4-4

Sol) Given to find the recursion tree upper bound on $T(n) = 2T(n-1) + 1$

* Rate of increasing in No. of subproblem = 2
* Rate of decrease in subproblem = 1

total cost of the tree

$$T(n) = \sum_{i=0}^{n} 2^i \cdot 1$$

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

$$= 2^{n+1} - 1$$

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

$$\leq c2^n$$

$$= O(2^n)$$

Using Substitution Verifing

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

$\leq 2(c2^{n-1} - b) + 1$

$\leq 2c2^{n-1} - 2b + 1$

$\leq c2^n - b$

$= O(2^n)$

4.5-4

Sol) Given recurrence, $a = 4$, $b = 2 \Rightarrow T(n) = 4T(n/2) + n^2 \lg n$

- Rate of increase of No. of subproblems = 4
- Rate of decrease of subproblem = 2.
- Total cost of level i

$$T(n) = \sum_{i=0}^{\lg n} 4^i \cdot c\left(\frac{n}{2^i}\right)^2 \cdot \lg\left(\frac{n}{2^i}\right)$$

$$= \sum_{i=0}^{\lg n} 4^i \cdot c\left(\frac{n^2}{2^{2i}}\right)(\lg n - \lg 2^i)$$

$$= cn^2 \sum_{i=0}^{\lg n} (\lg n - \lg 2^i)$$

$$= cn^2 \left(\sum_{i=0}^{\lg n} \lg n - \sum_{i=0}^{\lg n} \lg 2^i\right)$$

$$= cn^2 \left(\left(\frac{\lg n}{2}\right)^2 - \left(\frac{\lg n}{2}\right)\right)$$

$$\leq \frac{cn^2 (\lg n)^2}{2}$$

$$\leq cn^2 \lg^2 n$$

4.1 (Master's Theorem)

- We know that master's theorem Basically have 3 main rule to followed
- We have to find the question in which case it will fall.

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

1. if $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then
$$T(n) = O(n^{\log_b a})$$

2. if $f(n) = O(n^{\log_b a})$, Then $T(n) = O(n^{\log_b a} \lg n)$

3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficient large n, then $T(n) = O(f(n))$.

a) Here $a = 2$ and $b = 2$ So, $\log_b a = 1$ and
$f(n) = n^4 = n^{1+3} = \Omega(n^{\log_b a + \epsilon})$ for $\epsilon = 3$
∴ So from above Equation we got that case(3) need to be applied.
$af(n/b) = 2f(n/2) = n^4/8 \leq cn^4$
∴ from that we $T(n) = O(n^4)$

b) Given $a = 1$ and $b = 10/7$, So $\log_b a = \log 1 = 0$
and $f(n) = n^{0+1} = (n) = \Omega(n^{\log_b a + \epsilon})$ $k = 1$

$af(n/b) = f(7n/10) = 7n/10$ again case(3).

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

c) Given $a = 16$ and $b = 4$, So, $\log_b a = 2$ and
$f(n) = n^2 = O(n^{\log_b a})$ $\log_4 4^2 = 2(1)$
$= 2$

$f(n) = n^2 = O(n^{\log_b a})$ we need to apply (2)
$T(n) = O(n^2 \log n)$

d) Given $a=7$ and $b=3$ so, $\log_3 a = 1.77$ and $f(n) = n^2 = n^{1.77+0.23} = \Omega(n^{\log_b a+\epsilon})$

$\therefore af(n/b) = 7f(7n/3) = 7n^2/9 \le cn^2$

* Need to apply case(3).

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

e) Given $a=7$ and $b=2$ so, $\log a = \log 7 = 2.8$ and $f(n) = n^2 = n^{2.8-0.8} = \Omega(n^{\log_b a+\epsilon})$

* we Need to apply case(1) $T(n) = \Theta(n^{\log 7})$

f) Given $a=2$ and $b=4$. so $\log a = 1/2$ and $f(n) = \sqrt{n} = \Omega(n^{\log_b a})$

we can apply case(2) : $T(n) = \Theta(\sqrt{n}\lg n)$

g) Given $T(n) = T(n-2) + n^2$

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

$= T(n-6) + (n-4)^2 + (n-2)^2 + n^2$

$= T(0) + 2^2 + 4^2 + \cdots (n-4)^2 + (n-2)^2 + n^2$

$n/2$

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

$$= T(0) + \sum_{i=0}^{n/2} n^2 - \sum_{j=0}^{n/2} 4nj + \sum_{i=0}^{n/2} 4i^2$$

$= T(0) + \Theta(n^3) - 4(n^3) + \Theta(n^3)$

$= \Theta(n^3)$