Name — Aayush Dhaundiyal

Student Id — 20011324

Section — D

Class No — 01

University No — 2018054

Assignment – 1 TCS-409

Q 1 - A) `int a[n], b[100][n], c[n][n];`

`for(i=n; j=0; i>0; i/=2; j+=i)`

Sol i getting halved for each iteration

Initial value of $i = n$

After 1st loop $\Rightarrow i = n/2$

After 2nd loop $\Rightarrow i = n/2 \times \frac{1}{2} = n/2^2$

" 3rd loop $\Rightarrow i = n/2^3$

After k^{th} loop $i = n/2^k$

$$n/2^k = 1$$

$$n = 2^k$$

$$\log_2 n = k \log_2 2$$

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

Aayush Dhaundiyal
D-1

Time Complexity = O(log n)

No extra space used

∴ SC = O(1)

(B) int j, n;
j = 1;
while (j<=n)
j = j * 2;

Value of j is getting incremented twice

After 1st while loop j ⇒ 1
" 2nd while loop j = 2
∴ Kth " " j = 2^K

$n = 2^K$

Taking log both sides

$\log n = \log 2^K$

$\log n = K \log 2$

∴ TC = O(log n)
SC = O(1).

Number of comparisons = log n

Total Number of comparision = log n + 1.

Aayush Bhandiyal
D-1

© Fun()
{
$n = 2^{2^k}$

for (i = 1 to n)
  j = 2;
while (j <= n)
  j = j^2
}

For loop will run till i = n

∴ TC of outer loop = O(n)

For inner of while loop –

1st iteration $j = 2^{2^0} = 2$
2nd iteration $j = 2^{2^1} = 4$
3rd iteration $j = 2^{2^3} = 16$
k iteration $j = 2^{2^k}$

$n = 2^{2^k}$

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

$\log(\log n) = k \log 2$

∴ TC for inner loop = $O(\log(\log n))$

∴ Total TC of inner loop = $O(n) * O(\log(\log n))$

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

$SC = O(1)$

Aayush Chaundiyal
D-1

② Recursive relation

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

Solving using substitution method

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

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

$= T(2^{k/2}) + 2^k$

$= T(2^{k/4}) + 2^{k/2} + 2^k$

$= T(2^{k/2^3}) + 2^{k/2^2} + 2^{k/2}$

$= T(2^{k/2^m}) + 2^{k/2^{m-1}} \ldots 2^k$

$2^{k/2^m} = 2$

$\frac{k}{2^m} = 1 \Rightarrow k = 2^m = \log k$

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

Aayush Dhaundiya
D-1

Q2

$$f(n) = 4n + 100\sqrt{n}$$

(i) $g(n) = \log_2 n$

Using limits method we have :-

(1) $\lim_{n \to 0} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{4n + 100\sqrt{n}}{\log_2 n} \left(\frac{\infty}{\infty}\right)$

Using L-Hospital Rule

$$\lim_{n \to \infty} \frac{4 + 50\sqrt{n}}{1/n \times \log_2 e} = \frac{4+0}{0 \cdot \log_2 e} = \frac{4}{0}$$

$\Rightarrow f(n) > g(n)$

$\therefore f(n) = \Omega(g(n))$

Now $c \cdot g \leq f(n)$

let $c = 1, n_0 = 1 \Rightarrow 1 \cdot \log_2 1 \leq 4(1) + 100\sqrt{1}$

$$\boxed{\therefore c = 1, n_0 = 1}$$

(ii) $g(n) = \sqrt{n}$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{4n + 100\sqrt{n}}{\sqrt{n}} = \lim_{n \to \infty} 4\sqrt{n} + 100$$

$= \infty$

$\therefore f(n) > g(n)$

$\therefore f(n) = \Omega(g(n))$

$c \cdot g(n) \leq f(n)$

For $c=1, n_0=1$

$\Rightarrow 1.\sqrt{n} \leq 4 \times 1 + 100\sqrt{1}$

$\Rightarrow 1.\sqrt{1} \leq 4 + 100$

$\Rightarrow 1 \leq 104$, which is true

$\therefore c=1, n_0=1$

(iii) $g(n) = n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{4n + 100\sqrt{n}}{n} = \lim_{n\to\infty} 4 + \frac{100}{\sqrt{n}} = 4+0=4$$

$\therefore f(n) = g(n)$

$\therefore f(n) = O(g(n))$

for $c=104, n_0=1$

$104 \cdot n = 4(1) + 100\sqrt{1}$

$104 = 104$, which is true

$\therefore c=104, n_0=1$

(iv) $g(n) = n^2$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{4n + 100\sqrt{n}}{n^2} = \lim_{n\to\infty} \frac{4n}{n^2} + \frac{100\sqrt{n}}{n^2}$$

$$= 0$$

$\therefore f(n) < g(n)$

$\therefore f(n) = O(g(n))$

for $c = 105$, $n_0 = 1$

$105 \cdot (n^2) > 4 \times n + 100\sqrt{n}$

$105 > 104$ which is true.

2(B) $2n = O(n^2)$ or $2n = O(n^2)$

let $f(n) = 2n$, $g(n) = n^2$

$$\lim_{n\to\infty} \frac{2n}{n^2} = \lim_{n\to\infty} \frac{2}{n} = \frac{2}{\infty} = 0 \text{ (case of Big O)}$$

$\therefore f(n) > g(n)$ (for Big O) | $f(n) < g(n)$ (for small o)

$f(n) = o(g(n))$ | $f(n) = o(g(n))$.

let $c = 1$, $n_0 = 2$

$2 \times 2 \leq 2^2$ (Big O)

$4 \leq 4$ (true)

$c = 1$, $n_0 = 3$

$2(3) \leq 1 * 9$ (true)

$n_0 \geq 2$ & $c = 1$

$2 \times 2 < 2^2$ (small o)

$4 < 4$ (false)

let $c = 1$, $n_0 = 3$

$6 < 9$ (true)

This is true for all $n_0 > 2$ and $c = 1$

Aayush Chaundiyal
D-1

3(A) $f(x) = x^2 + 5$, $g(x) = 3x^2 + 4x$.

$$\Rightarrow \lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{x^2+5}{3x^2+4x} \quad \left(\frac{\infty}{\infty}\right)$$

$$= \lim_{x\to\infty} \frac{x^2(1+5/x^2)}{x^2(3+4/x)} = \lim_{x\to\infty} \frac{1+5/x^2}{3+4/x}$$

$$= \frac{1+0}{3+0} = 1/3 = (\text{constant}).$$

$\therefore f(x) = g(x)$

$f(x) = \Theta(g(n))$

3

(B) $f(x) = 2x + 1$, $g(x) = 3\log^2 x + 5$

$$\Rightarrow \lim_{x\to\infty} \frac{2x+1}{3\log^2 x + 5} = \left(\frac{\infty}{\infty}\right)$$

using L-Hospital Rule

$$\Rightarrow \lim_{x\to\infty} \frac{2}{6\log x/x} = \lim_{x\to\infty} \frac{2x}{6\log x} = \frac{2}{0} = \infty$$

$\therefore f(x) > g(x)$

$\Rightarrow f(x) = \Omega g(x))$

3

(C) $f(x) = \sqrt{x}$, $g(x) = 4\log_2 x$

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{\sqrt{x}}{4\log_2 x} = \left(\frac{\infty}{\infty}\right)$$

Aayush Chaudhary
D-1

Use L-Hospital Rule

$$\lim_{x\to\infty} \frac{1/2\sqrt{x}}{4/x \log e/2} \Rightarrow \lim_{x\to\infty} \frac{x \log e/2}{4 \cdot 2\sqrt{x}}$$

$$= \lim_{x\to\infty} \frac{\sqrt{x} \log e/2}{4} \to$$

$$= \infty .$$

3 (D) $f(x) = 2\log_2 2x$, $g(x) = \log_3 2x$

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{2\log_2 2x}{\log_3 2x} = \left(\frac{\infty}{\infty}\right)$$

Use L-Hospital Rule

$$\lim_{x\to\infty} \frac{2/x \log_2 e}{2/2x \log_3 e} \Rightarrow \lim_{x\to\infty} \frac{2\log_3 e}{\log_2 e} = \text{constant}.$$

$\therefore f(x) = g(x)$

$\Rightarrow f(x) = \Theta(g(x))$.

3(E) $f(x) = 3000 x^{3/2}$, $g(x) = \log_2 3x$, $h(x) = (\log x)^3$

Taking log of all functions

$f(x) = 3000\, x^{3/2}$

$= \log(3000\, x^{3/2}) = \frac{3}{2} \log(3000x)$

$g(x) = \log_2 3x$

$= \log(\log_2 3x)$

Aayush Dhaundiyal
0-1

$h(x) = (\log x)^3$

$= 3 \log(\log x)$.

For $f(x)$ & $g(x)$: let $x = 2^{10}$

$f(x) = 3/2 \times 10 \log_2(6000) = 130.3$

$g(x) = \log(\log 3x) = \log(\log 3 + \log x)$

$= 2.40$

$\therefore f(x) > g(x)$

$\Rightarrow f(x) = \Omega(g(x))$

for $f(x)$, (n), $x = 2^{10}$

$f(x) = 130.3$

$h(x) = 3 \cdot \log(\log 2^{10})$

$= 3 \cdot \log(10)$

$= 6.90$

$f(x) > h(x)$

$\therefore f(x) = \Omega(h(x))$

for $g(x) \to h(x)$; let $n = 2^{10}$

$g(x) = 2.40$, $h(x) = 6.90$

$g(x) < h(x)$

$\therefore g(x) = o(h(x))$

Aayush Chaudhary
0-1

Q4 (a) $f(n) = O(g(n))$

$f(n) = O(n)$

for $n = 1000$

Time $= R \times 1000 = \frac{1}{1000} \times 1000 = 1 \text{sec}$

for $n = 2000$

$T = R * 2000 = 2 \text{sec}$

for $n = 10,000$

$T = 100 \text{ sec}$

(b) $f(n) = O(n^2)$

Time $= R n^2$

$n = 1000$, Time $= 10 \text{ sec} \leftarrow$ given

$\Rightarrow R = \frac{\text{Time}}{n^2} = \frac{10}{(1000)^2} = \frac{1}{10^5}$

If $n = 2000$, Time $= \frac{1}{10^5} \times (2000)^2$

$= \frac{1}{100000} \times 2000 \times 2000$

$= 40 \text{ sec}$

If $n = 10,000$, Time $= \frac{1}{10^5} \times (10000)^2$

$= 1000 \text{ sec.}$

Aayush Dhaundiyal
0-1

(c) $f(n) = 0\sqrt{n}$

$n = 1000$

Time = 10 sec

Time $= k \cdot \sqrt{n}$

$$k = \frac{\text{Time}}{\sqrt{n}} = \frac{10}{\sqrt{1000}} = \frac{10}{31.62} = 0.3162$$

If $n = 2000$,

Time $= 0.1362 \times \sqrt{2000}$

$= 14.140$ sec

If $n = 1.0000$ Time $= 0.3162 \times \sqrt{10000}$

$= 31.62$ sec

(D) $f(n) = 0(\log(n))$

for $n = 1000$.

Time = 10 s

Time $= k \cdot \log_{10} n$

$$k = \frac{\text{time}}{\log n} = \frac{10}{\log 1000} = 10/3$$

If $n = 2000$,

$$\text{Time} = \frac{10}{3} \times 3.30$$

$= 10 \times 1.1$

$= 11$ s

Q5 Aayush Dhaundiyal
D-1

For A, $T(n) = 4n^{1.5}$

For B, $T(n) = 0.04n^{1.75}$

(A) for $n = 10^{6}$

$A = 4 \times 10^{6} \times 1.5$

$= 4 \times 10^{12}$

$B = 0.04 \times 10^{6} \times 1.75$

$= 0.04 \times 10^{14}$

$= 4 \times 10^{12}$

Both A and B software produce result in same time.

Both are better.

(B) $T(n) = 8T(n/32) + n \log n$

(B) for $n > 10^{8}$, let $n = 10^{15}$

$A = 4 \times 10^{15} \times 1.5$

$B = 0.04 \times 10^{15} \times 1.75$.

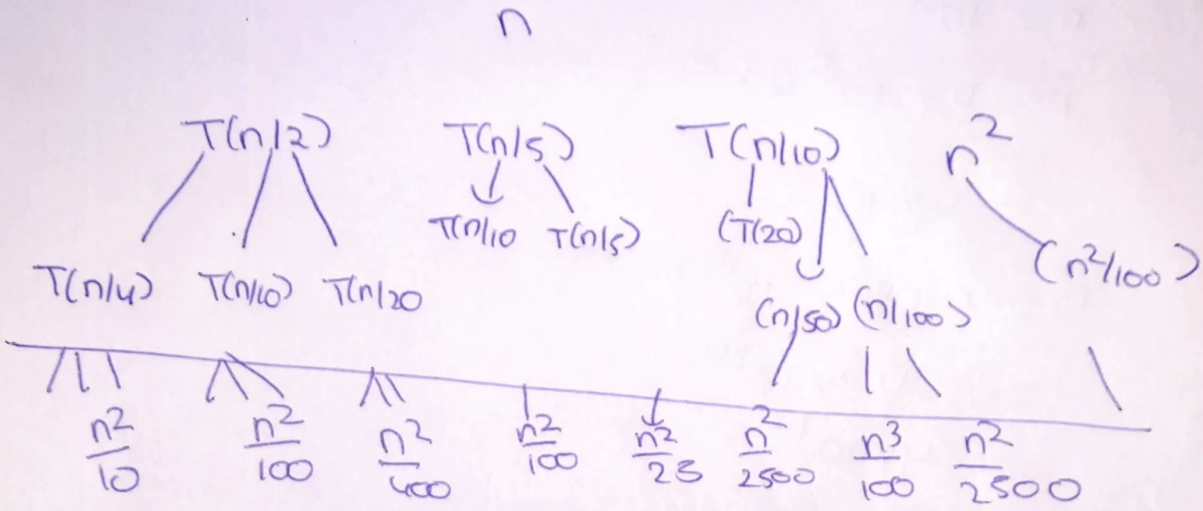
Software A performs better.

Aayush Dhaundiyal
Aayush Dhaundiyal
D - 1

Q6

A) $T(n) = T(n/2) + T(n/5) + T(n/10) + n^2$

Recursion Tree method



$$\left(n^2 + \frac{30n^2}{100} + \left(\frac{30}{100}\right)^2 n^2 + \dots\right)$$

decreasing GP $\Rightarrow 1$

$$n^2\left(1 + \frac{30}{100} + \left(\frac{30}{100}\right)^2 \dots\right)$$

$$n^2 \times 1 = n^2$$

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

Aayush Dhaundiyal
D—1

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

For eq of $aT(n/b) + f(n)$,
master theorem can be used

$a = 3$
$b = 3$
$R = \log_b a = \log_3 3$

$n^1 = n^1$
$n^R < f(n)$
$T(n) = O(n \log n)$

C) $T(n) = 2T(n-1) + n$ for $n >= 2$ $T(1) = 1$

Using substitution method

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

Put (I) in original eq.

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

find $T(n-2)$

$T(n-2) = 2^2 2^3 T(n-3) + n - 2$ — (III)

Put (II) in (III)

$T(n) = 2^2 \cdot 2^3 T(n-3) + (n-2) + (n-1) + n$.

after R steps.

$T(n) = 2^R T(n-k) + (n-k-1) + (n-k-2) \ldots n$

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

$k = n - 1$

Put $k = n-1$

$$T(n) = 2^{(n-1)} T(1) + (n-1) + (n-2) + (n-3) \cdots n-k$$

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

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

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

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

(5) $T(n) = 2T(n^{0.5}) + 1 \quad T(1) = 1$

assume $n = 2^k \Rightarrow \log_2 n = k \log_2 2 \Rightarrow \log_2 n = k.$

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

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

assume $T(2^k) = S(k)$

$$S(k) = 2S(k/2) + 1$$

using masters theorem

$a = 2, \; b = 2$

$$S(k) = k^{\log_2 2}$$

$$S(k) = O(k^1)$$

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

but $n = 2^k$ and $k = \log_2 n$

$$\boxed{Tn = O(\log n)}$$

Aayush Dhandiya

Q7 A $2^{1.4n}$, $3^{1.2n}$, n^{1000} , n^{1000} , 1.0001^{n}

Taking log of all

order of fastest to slowest is :-

$$n^{100} < n^{1000} < 1.0001^{n} < 2^{1.4n} < 3^{1.2n}$$

(B) $n\log n$, $n^2 \log n$, $n(\log n)^3$, $(\sqrt{n})^{2.3}$

$$\sqrt{n}^{2.3} < n\log n < n(\log n)^3 < n^2 \log n$$

Qs

Priori Analysis	Posteriori Analysis
(1) Absolute Analysis	(2) Relative analysis
(2) Independent of compiler	(2) Dependent on compiler and hardware.
(3) Gives approximate answer.	(3) Gives exact answer.
(4) Uses asymptotic notations.	(4) Does not uses asymptotic notations

Aayush Dhanotiya
D-1

Q9

Sol Exponential Search Technique is explicitly designed for unbounded array.

It contains two steps :-

→ It determines range.

→ Perform binary search on that range.

Q10

Sol Yes, Binary Search can be applied in sorted linked list.

TC- O(n)

Because of traversing the nodes again & again it result in linear search

Approach

- Start head : set to head and last to NULL
→ Using two pointers approach middle is calculated
→ Eg If middle data < value, move to upper half
→ Else go to lower half.

TC- O(n)
SP- O(1)