HOMEWORK 3 → SUBMITTED BY:

ARSHITHA BASAVARAJ

1.

a) $T(n) = 9T(n/3) + n^2$

Using master method,

$a=9, \ b=3, \ f(n)=n^2$

CASE I: $f(n)$ is $O(n^{\log_b a-\varepsilon})$ for some const. $\varepsilon > 0$

$n^{\log_b a-\varepsilon} = n^{(\log_3 9)-\varepsilon}$

$= n^{2-\varepsilon}$ Let's assume $\varepsilon = 0.5$

Scanned by CamScanner

$= n^{2-\varepsilon}$ Let's assume $\varepsilon = 0.5$

$\Rightarrow n^{\log_b a - \varepsilon} = n\sqrt{n}$

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

$f(n) \leq c \cdot g(n) \quad \forall\, n > n_0$

$n^2 \leq c \cdot g(n) \quad \forall\, n > n_0$

$n^2 \leq c \cdot n\sqrt{n} \quad \forall\, n > n_0$

There doesn't exist a 'c' for which $n^2 \leq c \cdot n$

Scanned by CamScanner

CASE 2 :

$f(n)$ is $\Theta(n^2)$

$\Rightarrow n^2$ is $\Theta(n^2)$

$\Rightarrow T(n)$ is $\Theta(n^2 \log n)$

b) $T(n) = T(n-1) + 20 \log n$

$T(1) = T(0) + 20 \log 1$

$T(2) = T(1) + 20 \log 2$

$= T(0) + 20 [\log 1 + \log 2]$

Scanned by CamScanner

$$T(3) = T(2) + 20 \log 3$$
$$= T(0) + 20[\log 1 + \log 2 + \log 3]$$

By the same logic,

$$T(n) = T(n-1) + 20 \log n$$

$$= T(0) + 20[\log 1 + \log 2 + \dots + \log n]$$

$$T(n) = T(0) + 20 \log n!$$

Scanned by CamScanner

Using Stirling's approx.

$$\cancel{T(n) = \Theta(1) + 20\left[n\log n - n + O(\log n)\right]}$$

$$\cancel{= \Theta(1) + 20(n\log}$$

$$T(n) = \Theta(1) + 20\left[n\ln n - n + O(\ln n)\right]$$

$$= \Theta(1) + 20\,\Theta(n\ln n)$$

$\therefore$ $T(n)$ is $\Theta(n\ln n)$

Scanned by CamScanner

c. $T(n) = 5T(n/2) + n\sqrt{n}$

Using Master method,

$a=5, \quad b=2, \quad f(n) = n\sqrt{n}$

$n^{\log_b a} = n^{\log_2 5}$

$\Rightarrow \quad n^{(\log_2 5)-\varepsilon} > n^{(\log_2 4)-\varepsilon}$

$n^{(\log_2 5)-\varepsilon} > n^{2-\varepsilon} \geq n\sqrt{n}$

If $\varepsilon = 0.5$,

then $n^{(\log_2 5)-0.5} > n\sqrt{n} \geq n\sqrt{n}$

Let $g(n) = n^{\log_2 5}$

$f(n) \leq c.g(n) \quad \forall\, n > n_0$

$n\sqrt{n} \leq c.\, n^{(\log_2 5)-0.5} \quad \forall\, n < n_0$

for $c = 1$ & $n_0 = 1$

$n\sqrt{n} \leq n^{(\log_2 5)-0.5}$

$\Rightarrow f(n)$ is $O(n^{\log_b a - \varepsilon})$

$\therefore$ By case 1 of master method,

$$T(n) \text{ is } \Theta\left(n^{\log_b a}\right)$$

$$\Rightarrow T(n) \text{ is } \Theta\left(n^{\log_2 5}\right)$$

d. $T(n) = aT(n/a) + n^2 \log n$, for constant $a > 1$

$a = a \quad b = a \quad f(n) = n^2 \log n$

$n^{\log_b a} = n^{\log_a a} = n$

$g(n) = n \;; \quad f(n) = n^2 \log n$

CASE: 1:$\Rightarrow$

$$f(n) \leq c.g(n) \quad \forall n > n_0$$

$$n^2 \log n \leq c.n \quad \forall n > n_0$$

There doesn't exist any 'c' s.t.,

$$n^2 \log n \leq c.n$$

Scanned by CamScanner

CASE 3:

$$g(n) \leq c.f(n) \quad \forall n > n_0$$

$$\cdot n \leq c.n^2 \log n \quad \forall n > n_0$$

$$n \leq n^2 \log n \quad \forall n > 1$$

$$c = 1$$

Also,

$$\Rightarrow \quad n^{1+\varepsilon} \leq n^2 \log n \quad \forall n > 1$$

$$\varepsilon\ c = 1$$

for $\varepsilon = 0.5$

$$n\sqrt{n} \leq n^2 \log n$$

$$\Rightarrow f(n) \text{ is } \Omega\left(n^{\log_b a + \varepsilon}\right) \rightarrow (1)$$

$$a \cdot f(n/b) = a \cdot \frac{n^2}{a^2} \log(n/a)$$

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

$$c \cdot f(n) = c \cdot n^2 \log n$$

$$\Rightarrow a f(n/b) \leq c \cdot f(n) \rightarrow (2)$$

Scanned by CamScanner

From ① & ②,

$T(n)$ is $\Theta(n^2 \log n)$

Scanned by CamScanner

e.) $T(n) = T(n-1) + n^5$

$T(1) = T(0) + 1^5$
$T(2) = T(0) + 1^5 + 2^5$
$T(3) = T(2) + 3^5$
$= T(0) + 1^5 + 2^5 + 3^5$

|||ly,
$T(n) = T(n-1) + n^5$
$= T(0) + [1^5 + 2^5 + 3^5 + \dots + n^5] \rightarrow (1)$

According to Wolfram Alpha,

$$\sum_{k=1}^{n} k^5 = \frac{1}{12} n^2 (n+1)^2 (2n^2 + 2n - 1) \rightarrow (2)$$

Substituting (2) in (1)
$\Rightarrow$

$$T(n) = T(0) + \frac{1}{12} n^2 (n+1)^2 (2n^2 + 2n - 1)$$

$T(n) = T(0) + \Theta(n^6)$ [on simplification]

$\therefore T(n)$ is $\Theta(n^6)$

Scanned by CamScanner

2a. There is no recurrence in this case.

```
void A (int n) {
  for (int i = 0; i < n; i++)            => n times
      for (int j = 1; j < n; j = j*2)    => ⌈n/2⌉ times
           print (i*j);
}
```

$$\Rightarrow \Theta\left(n \times \frac{n}{2}\right) = \Theta\left(\frac{n^2}{2}\right)$$

$\Rightarrow$ worst-case

Scanned by CamScanner

b. void B (int A[1...n], int i){
    if (i==(n+1))
        print(A);   } const. time
    else
    for (int j=i; j<=n ;j++){
        swap (A,i,j);
        B (A, i+1);
        swap (A,i,j);
    }
}

Recursion formula: $T(n) = (n-1)\,T(n-1) + 2\,\Theta(n)$

Scanned by CamScanner

$$T(n) = (n-1)\left[(n-2)\, T(n-2) + 2\,\Theta(n)\right] + 2\,\Theta(n)$$

$$\Rightarrow T(n) = (n-1)(n-2)\ldots 1(T(1)) + \Theta(n)$$

$$\Rightarrow T(n) = n!\, T(n) + \Theta(n)$$

$\therefore$ $T(n)$ is $\Theta(n!)$ $\Rightarrow$ worst case.

Scanned by CamScanner

c. int c (int n) {
    int accumulator = 0;
    if (n == 0)
        return 1;
    else {
        for (int i = 0; i < n; i += 2) ⇒ $n/2$ times
            for (int j = 0; j < n; j += 2) ⇒ $n/2$ times
                accumulator++;
    }
    return accumulator + c(n/3) + c(n/3) + c(n/3)
}

Recursion formula:

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

$a = 3$, $b = 3$, $f(n) = n^2/4$

$$n^{\log_b a} = n^{\log_3 3} = n$$

CASE 3: $n^{1+\varepsilon} \leq c \cdot \dfrac{n^2}{4} \quad \forall n > n_0$

Let $\varepsilon = 0.5$

Scanned by CamScanner

then

$$n^{1.5} \leq c.\frac{n^2}{4} \quad \forall\ n > n_0$$

$$n\sqrt{n} \leq \frac{n^2}{4} \quad \text{for } c=1,\ n_0 = 1$$

$$\forall\ n > n_0$$

$\Rightarrow f(n)$ is $\Omega(n^{\log_b a+\varepsilon})$

a. $f(n/b) = 3f(n/3) = \cancel{3}\times\frac{n^2}{3\cancel{9}\times 4} = \frac{n^2}{12}$

c. $f(n) = c\frac{n^2}{4}$ [for $c = 0.9$]

$\Rightarrow af(n/b) \leq c.f(n)$

$\therefore T(n)$ is $\Theta(n^2/4)$

or $T(n)$ is $\Theta(n^2)$

Scanned by CamScanner

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

$T_1(n) = 2T(n/2) + n$

$T_2(n) = 2T(n/3) + n$

$$T_2(n) < T(n) < T_1(n)$$

Using Master method on $T_1(n)$ & $T_2(n$

Using Master method on $T_1(n)$ & $T_2(n)$:

$T_1(n) = 2T(n/2) + n$

$a = 2, \quad b = 2, \quad f(n) = n$

$n^{(\log_b a) - \varepsilon} = n^{1-\varepsilon} = (n)^{0.5} = \sqrt{n}$

for $\varepsilon = 0.5$

$n \leq c.\sqrt{n} \quad \forall n > n_0 \rightarrow (1)$

There's no 'c' for which (1) is true.

Scanned by CamScanner

CASE 2: $n^{\log_a b} = n$

$\Rightarrow f(n)$ is $\Theta(n)$

$\Rightarrow f(n)$ is $\Theta(n^{\log_b a})$

$\Rightarrow T_1(n)$ is $\Theta(n \log n)$

Scanned by CamScanner

$$T_2(n) = 2T\left(\frac{n}{3}\right) + n$$

$a = 2$; $b = 3$; $f(n) = n$

$\log_3 2 = 0.6309$ (approx.)

$$n^{\log_3 2} = n^{0.6309}$$

Scanned by CamScanner

$n^{1.1309} \leq c \cdot n$

$f(n) = n$

$g(n) = n^{0.6309 + \varepsilon}$

if $\varepsilon = 0.37$

$g(n) = n^{0.6309 + 0.37}$

$\simeq n$

Scanned by CamScanner

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

$$n \leq c \cdot n \quad \forall n > n_0$$

$$c = 1/2 \qquad n_0 = 1$$

$$n \leq n/2 \quad \forall n > 1$$

$$\Rightarrow n \text{ is } \Omega(n)$$

$$a \cdot f(n/b) = 2 \times n/3$$

$$c f(n) = c \cdot n$$

$\therefore$ for $c = 0.9$

$$\underline{a f(n/b) \leq 0.9\, n}$$

$$\Rightarrow T_2(n) \text{ is } \Theta(n)$$

$\Rightarrow T(n)$ has the time complexity between $\Theta(n)$ and $\Theta(n \log n)$

Scanned by CamScanner

2d)

```
int D (int n){
  if (n<=1)
    return 1;
  else if (D(1)<10)
    return 45*D(n-2);
  else
    return 45;
}
```

Recursion formula:

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

$$T(n) = T(n-2) + \cancel{\text{const}}\ \Theta(1)$$

$$= T(n-4) + 2\Theta(1)$$

$$\Rightarrow T(n) = T(0) + \left\lfloor \frac{n}{2} \right\rfloor \Theta(1)$$

$$\therefore T(n) \text{ is } \Theta(n)$$

Scanned by CamScanner