Name: Ahmed El-Metwally Mohamed

Notes:

1- Read and write ➔ t1
2- Assign ➔ t2
3- Arithmetic Operation ➔ t3

Question 1:-

Algorithm 1: What is the Complexity of the following Algorithms?

```
int t = 0 ;
for ( int i = 1 ; i <= n ; i++)
        for ( int j = 0 ; j ∗ j < 4 ∗ n ; j++)
                for ( int k = 1 ; k ∗ k <= 9 ∗ n ; k++)
                        t++;
```

Solution

(int i = 1 ; i <= n ; i++) ➔ i = n

(int j = 0 ; j ∗ j < 4 ∗ n ; j++) ➔ j = $2\sqrt{n}$

(int k = 1 ; k ∗ k <= 9 ∗ n ; k++) ➔ k = $3\sqrt{n}$

So the time complexity = n * $2\sqrt{n}$ * $3\sqrt{n}$ = 6n^2 ➔ O(n^2)

Algorithm 2: What is the Complexity of the following Algorithms?

```
int z = 0 ;                              ➔ t2
int x = 0 ;                              ➔ t2
for ( int i = 1 ; i <= n ; i = i ∗ 3)    ➔ i = log3 n
{
        z = z + 5 ;                      ➔ t2 + t3
        z++;                             ➔ t2 + t3
        x = 2 ∗ x ;                      ➔ t2 + t3
}
```

Solution: so the time complexity = 1 *1 *2 *2*2 * $\log_3 n$ ➔ O(log n)

Algorithm 3: What is the Complexity of the following Algorithms?

int x = 0 ; ➔ t1
for (int i = 1 ; i <= n ; i = i ∗ 3) ➔ i = $\log_3 n$
{
    if (i % 2 != 0) ➔ t4 +t3
        for (int j = 0 ; j < i ; j++) ➔ j = n
            x++; ➔ t2 + t3
}

Solution: Time Complexity = 1*2*2* n* $\log_3 n$ ➔ O(n*log n)

Algorithm 4: What is the Complexity of the following Algorithms?

int fun (int n)
{
    int count = 0 ; ➔ t1
    for (int i = n ; i > 0 ; i /= 2) ➔ i = $\log_{1/2} n$
    for (int j = 0 ; j < i ; j++) ➔ j = n
    count += 1 ; ➔ t2+t3
    return count ;
}

Solution: Time Complexity = 1*2 n* $\log_{1/2} n$ ➔ O(n*log n)

Algorithm 5: What is the Complexity of the following Algorithms?

int n , rev ;
rev = 0 ; ➔ t1
while (n > 0) ➔ $\log_{1/10} n$
    {
    rev = rev ∗ 10 + n % 1 0 ; ➔ t2 + 3t3
    n = n / 1 0 ; ➔ t2 + t3
    }

Solution: Time Complexity = 1*4*2* $\log_{1/10} n$ ➔ O(log n)

Algorithm 6: What is the Complexity of the following Algorithms?

```
int fun1 ( int n)
{
int i , j , k , p , q = 0 ;                → t2
for ( i = 1 ; i < n ; ++i )                → i = n
{
        p = 0 ;                            → t2
        for ( j = n ; j > 1 ; j = j / 2)   → log1/2 n
        ++p ;                              → t2 + t3
        for ( k = 1 ; k < p ; k = k ∗ 2)   → log2 n
        ++q ;                              → t2 + t3
        }
        return q ;
}
```

Solution: Time Complexity = n *($\log_{1/2} n$ + $\log_2 n$) → O(n*log n)

Algorithm 7: What is the Complexity of the following Algorithms?

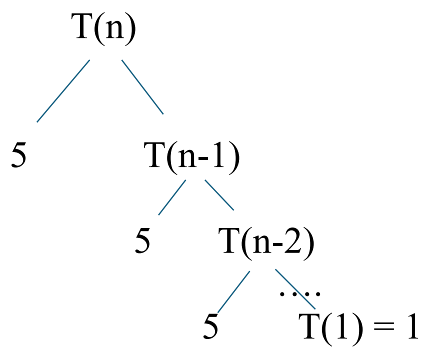
```
int i = 1 , z = 0 ;                        → 2t2
while ( z < n ∗ (n + 1) / 2)               → z= (0.5n^2 + 0.5n) → n
{
        z += i ;                           → t2 + t3
        i++;                               → t2 + t3
}
```

Solution: Time Complexity = O(n)

Question 2: Solve the following recurrence relations (use Recurrence Tree)

1- $T(n) = T(n-1) + 5, T(1) = 1$ and $n > 1$.

T(n) =

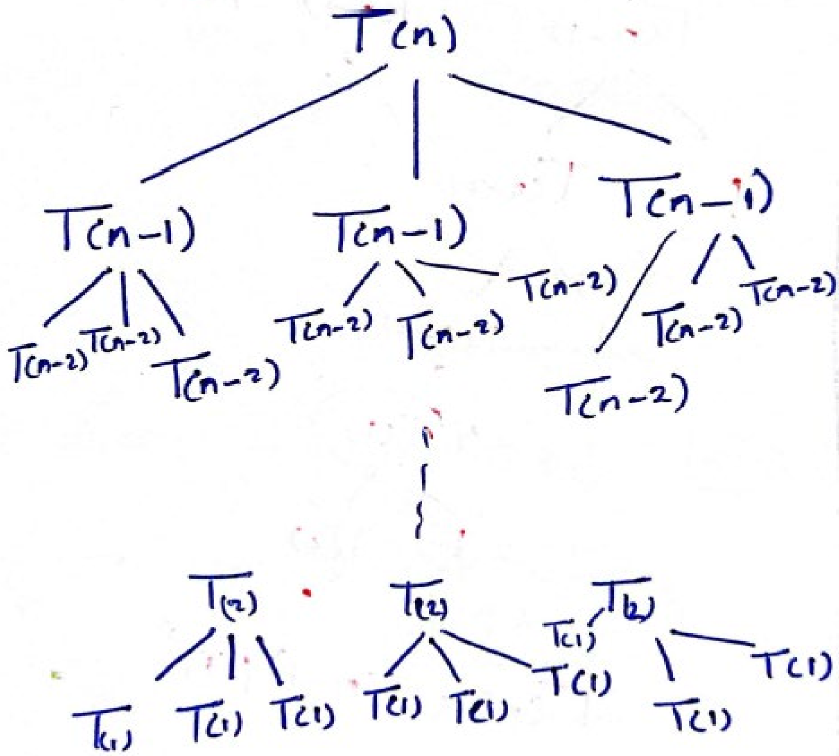


sol: O(n)

2- T(n) = 3T(n − 1),T(1) = 4 and n > 1.

$\boxed{2}$ $T(n) = 3T(n-1)$

$T(1) = 4 \quad n > 1$



Complexity $\longrightarrow O(3^n)$

3- T(n) = T(n/2) + n,T(1) = 1 and n > 1.

[3] $T(n) = T(n/2) + n$

$$\boxed{T(1) = 1 \quad n > 1}$$

$$\boxed{a = 1 \quad b = 2} \quad f(n) = n$$

عايزين نوصل n

$f(n) = n^K (\log n^P)$

$$n = n^1 [\text{Log } n]^0$$

$$\boxed{\begin{matrix} P = 0 \\ K = 1 \end{matrix}}$$

$$\text{Log}_b^a = \text{Log}_2^1 = 0$$

if $\text{Log}_b^a = 0 < K$

$$\boxed{0 < 1}$$

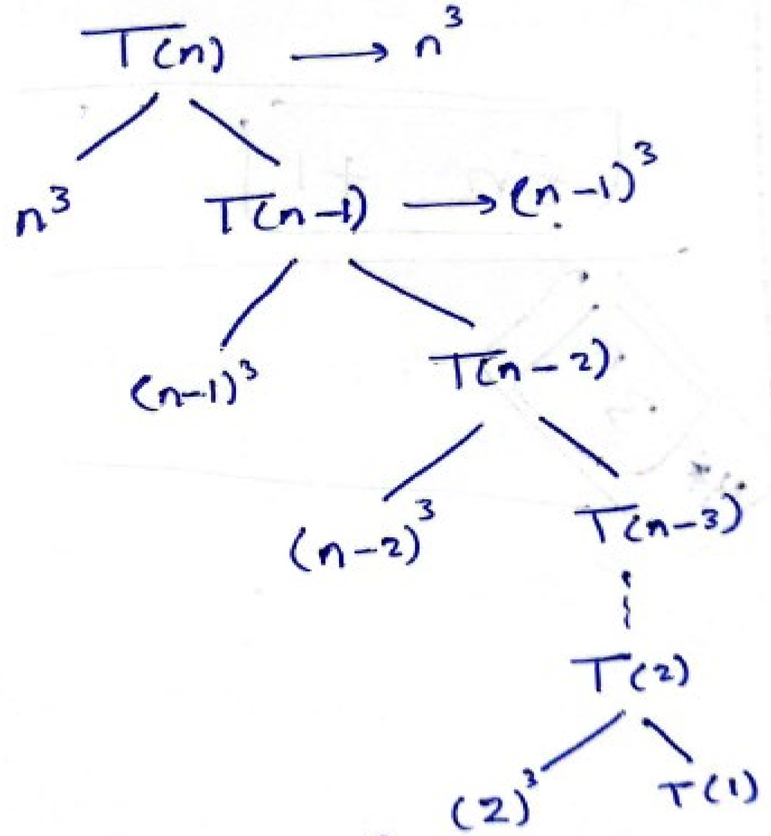
find P $\longrightarrow$ $P = 0$

$\therefore O(n^K (\text{Log } n^P))$

$\longrightarrow O(n \text{Log} n)$

4- T(n) = T(n − 1) + n3,T(1) = 1 and n > 1.

$\boxed{4}$ $T(n) = T(n-1) + n^3$



$T(n) = n^3 + (n-1)^3 + \cdots + 1$

متتابعة هندسية

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

Complexity $\rightarrow O(n^4)$

5- T(n) = 4T(n/3) + n,T(1) = 1 and n > 1.

---

[5] $T(n) = 4T(n/3) + n$

$a = 4 \quad b = 3 \quad f(n) = n$

$n^1 [\text{Log } n]^0 \qquad p = 0$

$K = 1$

$\text{Log}_b^a = \text{Log}_3^4 = 1.26$

$1.26 > K$

$\boxed{1.26 > 1}$

$\therefore F(n) \longrightarrow$ ~~[illegible]~~ $)$

$\longrightarrow O(n^{\text{Log}_3^4})$

6- T(n) = 2T(n/2) + log n,T(1) = 1 and n > 1.

$\boxed{6}$ $T(n) = 2T(n/2) + \text{Log } n$

$a = 2 \quad b = 2 \quad n = \text{Log } n$

$\text{Log } n \longrightarrow n^K [\text{Log } n]^P$

$n^0 [\text{Log } n]^1$

$\boxed{\begin{array}{l} P = 1 \\ K = 0 \end{array}}$

$\text{Log}_b^a = \text{Log}_2^2 = 1$

$1 > K \longrightarrow 1 > 0$

$\therefore f(n) \longrightarrow O(n^{\text{Log}_2^2})$

$= O(n)$

7- T(n) = 2T(n/2) + n2,T(1) = 1 and n > 1.

$\boxed{7}$ $T(n) = 2T(n/2) + n^2$

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

$n^2 \longrightarrow n^K [\text{Log}\, n]^P$

$n^2 [\text{Log}\, n]^0$ $\left[\begin{array}{l} K = 2 \\ P = 0 \end{array}\right]$

$\text{Log}_b^a = \text{Log}_2^2 = 1$

$1 < K$

$1 < 2$

$\therefore P = 0$

$\therefore \cancel{f(n) =} \hookrightarrow O(n^2)$

8- T(n) = 2T(n/4) +√n,T(1) = 1 and n > 1.

$\boxed{8}$ $T(n) = 2T(n/4) + \sqrt{n}$

$a = 2$ $b = 4$ $f(n) = n^{\frac{1}{2}}$

$n^{\frac{1}{2}} \longrightarrow n^{K} [\text{Log}\, n]^{P}$

$\longrightarrow n^{1/2} [\text{Log}\, n]^{0}$

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

$P = 0$

$\text{Log}_{4}^{2} = 2$

$2 < K \longrightarrow 2 < \frac{1}{2}$

$\therefore \longrightarrow O(n^{\log_b^a}) = O(n^2)$