Q.5) a.

// $P(x)$ is a polynomial of degree $d = 3^k - 1$ for some known k.

// let 'n' be the number of terms of the polynomial $P(x)$

// so here $n = d+1$

// def Eval (P, y, n): evaluates $P(y)$, given degree $(n-1)$ polynomial $P(x)$ and a number y:

def Eval (P, y, n):

$$P_1(x) = a_0 + a_3 x + \cdots + a_{n-3} x^{\lfloor \frac{n-1}{3} \rfloor}$$

$$P_2(x) = a_1 + a_4 x + \cdots + a_{n-2} x^{\lfloor \frac{n-1}{3} \rfloor}$$

$$P_3(x) = a_2 + a_5 x + \cdots + a_{n-1} x^{\lfloor \frac{n-1}{3} \rfloor}$$

$r_1 = \text{Eval}(P_1, y^3, n/3)$

$r_2 = \text{Eval}(P_2, y^3, n/3)$

$r_3 = \text{Eval}(P_3, y^3, n/3)$

return $(r_1 + r_2 * y + r_3 * y^2)$

Analysis

let $T(n)$ be the time complexity to compute Eval

$$T(n) = 3T(n/3) + O(n)$$

this can't be a constant as, to compute addition / multiplication of large numbers $O(n)$ time might be required in worst case.

using master's theorem
we get $T(n) = O(n\log n)$

as $n = d+1 \Rightarrow$ time complexity of this algorithm is $O(d \log d)$

Explaination :

Firstly, we are dividing the polynomial $P(x)$ into three equal parts based on remainder of 3, i.e.,

$A_1(y)$ consist of terms $a_i x^i$ where $(i \% 3) = 0$

$A_2(y)$ consist of terms $a_i x^i$ where $(i \% 3) = 1$

$A_3(y)$ consist of terms $a_i x^i$ where $(i \% 3) = 2$

The resultant polynomial will be,

$$P(y) = A_1(y) + A_2(y) + A_3(y)$$

To reduce the computational complexity, A_1, A_2, A_3 are modified as follows:

$$\left.\begin{aligned} A_1(y) &= P_1(y^3) \\ A_2(y) &= y\, P_2(y^3) \\ A_3(y) &= y^2\, P_3(y^3) \end{aligned}\right\}$$

i.e., by taking common y's we try to reduce the power of each of the 3 subproblems.

So now, $P(y) = P_1(y^3) + y P_2(y^3) + y^2 P_3(y^3)$

we further take the three subproblems and recursively call Eval on them to get r_1, r_2, r_3. Now using this three received terms we can simply compute $P(y) = r_1 + r_2 y + r_3 y^2$

Taking 3 parts on the basis of remainder of 3, helped in dividing the problem in 3 equal parts. Further after division we reduced the power of by taking y common and computing $P(y^3)$, so that time complexity reduces.

Q.5) b.

// $P(x)$ is a polynomial of degree $d = 3^k - 1$ for some known k.

// let n be the number of terms of $P(x)$

// so here $n = d+1$ and $P_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1}$

// $FFT_n(\langle a_0, a_1, \dots, a_{n-1} \rangle)$: it will compute the DFT of $P(x)$,

i.e., the values $\{ P(y) : y$ is the $(n-1)$th root of unity $\}$

```
FFT_n(<a_0, a_1, ..., a_{n-1}>)
    if n = 1 then return <a_0>
    else
        ω_n ← e^{2πi/n}
        ω ← 1
// here r0 % 3 = 0    <y_0^{r0}, ..., y_{n/3-1}^{r0}> ← FFT_{n/3}(<a_0, a_3, ..., a_{n-3}>)
// here r1 % 3 = 0    <y_0^{r1}, ..., y_{n/3-1}^{r1}> ← FFT_{n/3}(<a_1, a_4, ..., a_{n-2}>)
// here r2 % 3 = 0    <y_0^{r2}, ..., y_{n/3-1}^{r2}> ← FFT_{n/3}(<a_2, a_5, ..., a_{n-1}>)
        for k ← 0 to n/3 - 1 do
            y_k ← y_k^{r0} + ω y^{r1} + ω^2 y^{r2}          // here r0 % 3 = 0
            a ← ω * ω_n^3
            y_{k+n/3} ← y_k^0 + a y^{r1} + a^2 y^{r2}        // here r1 % 3 = 1
            b ← a * ω_n^3
            y_{k+2n/3} ← y_k^{r0} + b y_k^{r1} + b^2 y_k^{r2}  // here r2 % 3 = 2
            ω ← ω ω_n
    return <y_0, ..., y_{n-1}>
```

Analysis: Let $T(n)$ be the time complexity of FFT_n

$T(n) = 3T(n/3) + O(n)$ → Can't be constant time as addition or multiplication of large numbers can aid in linear time complexity.

using master's theorem

$T(n) = O(n \log n) \cong O(d \log d)$

Explaination

Base case : If $n=1$ i.e., FFT_1 then simply return the first coefficient as no more terms are present.

Here we are considering $n=9$ and computing FFT_9, to find out 9th roots of unity, i.e., for a given polynomial P, we will have $\langle A_0(\omega_9^0), A_1(\omega_9^1), \ldots, A(\omega_9^8) \rangle$.

We are dividing the polynomial $P(x)$ into three equal parts based on remainder of 3, i.e.,

$\forall a_i$ in $P_1(x)$ $i \% 3 = 0$

$\forall a_i$ in terms of $P_2(x)$ $i \% 3 = 1$

$\forall a_i$ in terms of $P_3(x)$ $i \% 3 = 2$

So now, $P(y) = P_1(y^3) + y P_2(y^3) + y^2 P_3(y^3)$

Thus using the above expression, the $A_i(\omega_9^i)$ terms are computed as illustrated below :

	P_1	P_2	P_3	$P(y)$
1. $A_0(\omega_9^0)$	$P_1(\omega_9^0)$	$P_2(\omega_9^0)$	$P_3(\omega_9^0)$	$P_1 + \omega_9^0 P_2 + \omega_9^0 P_3$
2. $A_1(\omega_9^1)$	$P_1(\omega_9^3)$	$P_2(\omega_9^3)$	$P_3(\omega_9^3)$	$P_1 + \omega_9^1 P_2 + \omega_9^2 P_3$
3. $A_2(\omega_9^2)$	$P_1(\omega_9^6)$	$P_2(\omega_9^6)$	$P_3(\omega_9^6)$	$P_1 + \omega_9^2 P_2 + \omega_9^4 P_3$
4. $A_3(\omega_9^3)$	$P_1(\omega_9^9)$	$P_2(\omega_9^9)$	$P_3(\omega_9^9)$	$P_1 + \omega_9^3 P_2 + \omega_9^6 P_3$
5. $A_4(\omega_9^4)$	$P_1(\omega_9^{12})$	$P_2(\omega_9^{12})$	$P_3(\omega_9^{12})$	$P_1 + \omega_9^4 P_2 + \omega_9^8 P_3$
6. $A_5(\omega_9^5)$	$P_1(\omega_9^{15})$	$P_2(\omega_9^{15})$	$P_3(\omega_9^{15})$	$P_1 + \omega_9^5 P_2 + \omega_9^{10} P_3$
7. $A_6(\omega_9^6)$	$P_1(\omega_9^{18})$	$P_2(\omega_9^{18})$	$P_3(\omega_9^{18})$	$P_1 + \omega_9^6 P_2 + \omega_9^{12} P_3$
8. $A_7(\omega_9^7)$	$P_1(\omega_9^{21})$	$P_2(\omega_9^{21})$	$P_3(\omega_9^{21})$	$P_1 + \omega_9^7 P_2 + \omega_9^{14} P_3$
9. $A(\omega_9^8)$	$P_1(\omega_9^{24})$	$P_2(\omega_9^{24})$	$P_3(\omega_9^{24})$	$P_1 + \omega_9^8 P_2 + \omega_9^{16} P_3$

P_1, P_2, P_3 values of row1 will reduce to $P_i(\omega_3^0)$

P_i values of row2 will reduce to $P_i(\omega_3^1)$

P_i values of row3 will reduce to $P_i(\omega_3^2)$

P_i of Row4 and Row7 will be reduced same as Row1. (of above table)

P_i of Row5 and Row8 will be reduced same as Row2. "

P_i of Row6 and Row9 will be reduced same as Row3. "

Thus the line 6,7,8 of the code is calculating $FFT_{n/3}$ for the three subproblems.

As mentioned above now we need to combine all the three subproblem's values to get $P(\omega_9^i)\ \forall i \in [0,8]$.

Also the multiplying term of P_2, P_3 of the Table, will overlap as follows:-

$\cancel{P_1 + \omega_9^0 P_2 + \omega_9^0 P_3 = P_1 + \omega_9^3 P_2 + \omega_9^3 P_3 = P_1 + \omega_9^6 P_2 + \omega_9^6 P_3}$

$\cancel{P_1 + \omega_9^1 P_2 + \omega_9^2 P_3 = P_1 + \omega_9^4 P_2 + \omega^8}$

$\cancel{\omega_9^0 = \omega_9^3 = \omega_9^6 \quad ; \quad \omega}$

in line 1,4,7 of table

$$\underset{(1)}{\underline{\omega_9^0}} = \omega_9^3 * \omega_9^0 = \underset{(4)}{\underline{\omega_9^3}} = \omega_9^3 * \omega_9^3 = \underset{(7)}{\underline{\omega_9^6}}$$

The similar is observed for subproblems 2,5,8 and 3,6,9.

Thus in line 11, 13 of the algorithm we are multiplying ω_n^3.

At the end, we get all the coefficients and then return.