

Day 3 - DSA

```
for(int i=1 ; i<n ; i++)  
{  
    —  
}
```

$$\begin{array}{l}
 \text{Time Complexity} \\
 i^2 = n \\
 i = \sqrt{n} \\
 i = n^{1/2} \\
 = \sqrt{n}
 \end{array}
 \quad
 \begin{array}{l}
 n=10 \\
 \boxed{i = 1 \quad 1^2 < 10} \\
 = 2 \quad 2^2 < 10 \\
 = 3 \quad 9 < 10 \\
 = 4 \quad 16 < 10 \\
 \times
 \end{array}$$

③ int i=1
while(i<n)
{
 i*=3
}

$\left\{ \begin{array}{l} \text{for } (i=1; i < n; i = i * 3) \\ \{ \\ \} \end{array} \right\}$

Time Complexity = $O(\log_3 n)$

④ `int i=10
while(i>0)
{ i /= 2 ⇒ {
 print()
}` for(int i=n ; i>0 ; i = i/2) O($\log_2 n$)

① LeetCode Pbm No :- 509

given:

Given:-

$$\text{Fib}(0) = 0 \quad \text{Fib}(1) = 1$$

* $\boxed{\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)}$

$$n=5?$$

$$2^2 =$$

$$3^2 = 8$$

Fibonacci Numbers

$$2^0 = 1$$

2

1

1

116 (2)

\longrightarrow Fib(n) Fib(5)

$$\text{Fib}(4) \leftarrow + \quad \text{Fib}(3) =$$

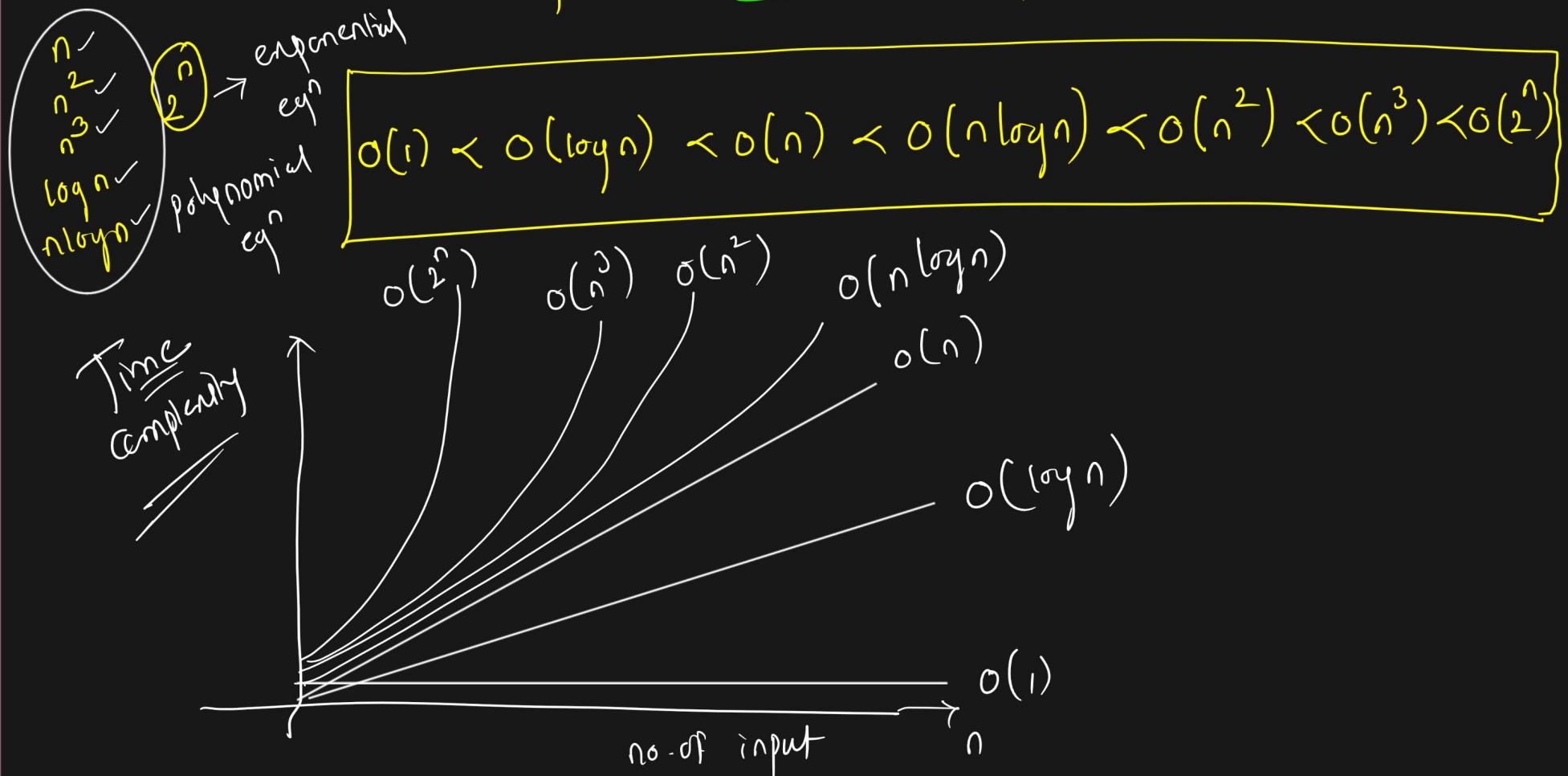
$$+ \underline{\text{Fib}(2)} \leftarrow \text{Fib}(2)$$

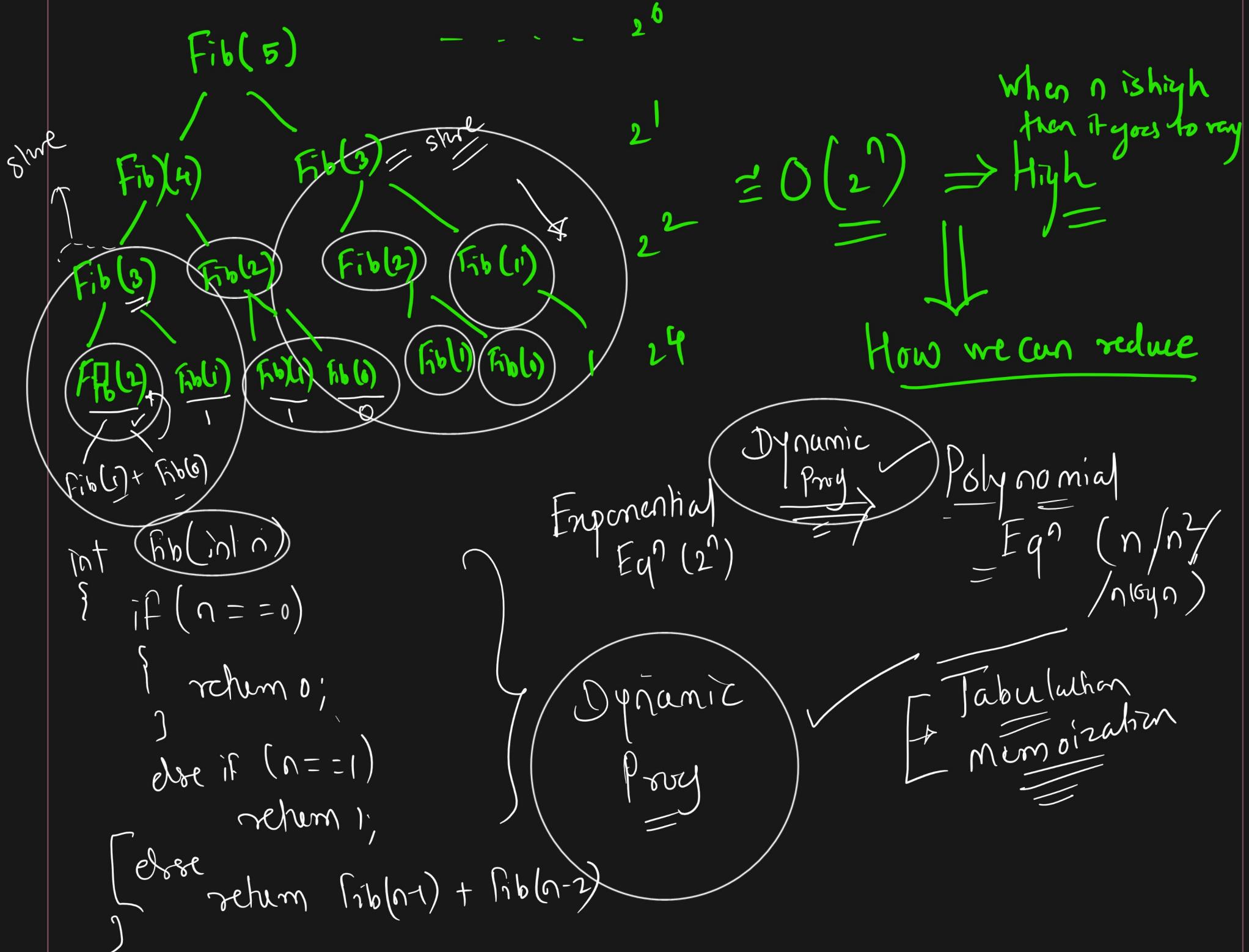
$$Fib(1) + Fib(0) \quad Fib(1) + Fib(0)$$

$$\begin{array}{r}
 2^4 = 16 \\
 \text{Fib}(1) + \text{Fib}(0) \\
 | \\
 1 + 0 \\
 | + 0 + 1 \\
 \hline
 2^5 = 20
 \end{array}$$

$$0, 1, 1, 2, 3, 5, 8, 13, \dots$$

$$\text{Time Complexity} = \underline{\underline{O(2^n)}} = \text{Exponential Eq?}$$





② Lecode from No: 231 :- Power of Two

$$n \Rightarrow 2^n = \underline{\underline{?}}$$

$$4 \Rightarrow 2^2 = 4 \checkmark \text{True}$$

$$\begin{matrix} n=3 \\ \therefore \end{matrix} \Rightarrow \begin{matrix} 2^2 = 4 \\ 2^3 = 8 \end{matrix} \text{False} \quad \underline{\underline{?}}$$

$\frac{16}{2} = \underline{\underline{?}}$
isPowerOfTwo (int n)

Bare cond
 {
 if ($n \geq 1$)
 return true;
 else if ($n < 1$)
 return false;
 } \Rightarrow else
 return isPowerOfTwo ($\frac{n}{2.0}$)
 $\underline{\underline{?}}$

$$\left. \begin{array}{l} n = 4 / .2 = \frac{4}{2} = \text{remainder } 0 \\ = 2 / 2 = 1 \rightarrow \text{True} \end{array} \right\}$$

$$\left. \begin{array}{l} n = 6 / 2 = \frac{3}{2} = 1 \frac{1}{2} = 0.75 \\ \downarrow 2^3 \end{array} \right\} \text{False}$$

$$n = 8 = 8 / 2 = 4 / 2 = 2 / 2 = 1, \text{True}$$

$$n = 9 = 9 / 2 = 4 \frac{1}{2} = 2 \frac{1}{2} = \frac{1.25}{2} = 0.625 \quad \underline{\underline{?}} \quad \text{False}$$

$$\frac{1}{2.0} = \frac{2.0}{2} = \frac{2}{2} = 1 \quad \underline{\underline{?}}$$

$$\frac{16}{2.0} = \frac{8.0}{2} = \left(\frac{8}{2} \right) \left(\frac{4}{2} \right) = \left(\frac{2}{2} \right) = 1 \quad \underline{\underline{?}}$$

int c=0;

while (n >= 5)

{

$$n = n / 5$$

$$c = c + n;$$

return c;

$$n = 25$$

while (n >= 5)

↓ {

$$n = n / 5 = 25 / 5 = 5$$

$$c = c + n = 0 + 5 = 5$$

}

return c;

$$\underline{\underline{n = 10}}$$

$$\underline{\underline{2}} = 5 \times$$

while (n >= 5) ←

{

$$n = 10 / 5 = 2$$

$$c = 0 + 2 = 2$$

}

return c;

$$10 / 5 = 2$$

while (n >= 5)

{

$$n = 5 / 5 = 1$$

$$c = c + n = 1 + 1 = 2$$

}

return c;

H.W. → Leetcode → 33 04