

Day 3 - DSA

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

Time Complexity

$$i^2 = n$$

$$i = \sqrt{n}$$

$$i = n^{1/2}$$

$$= \sqrt{n}$$

$n=10$

$$\left[\begin{array}{l} n=1 \quad 1^2 < 10 \\ \quad \quad = 2 \quad 2^2 < 10 \\ \quad \quad = 3 \quad 3^2 < 10 \\ \quad \quad = 4 \quad 4^2 < 10 \\ \quad \quad \quad \times \end{array} \right.$$

Time Complexity = $O(\sqrt{n})$

②

```
for(int i=1; i <= n; i++) → n times
{
    for(int j=1; j*j <= n; j++) → j^2 = n
    {
        j = sqrt(n)
    }
}
```

Total Complexity

$$= n \times \sqrt{n}$$

$$= n^1 \times n^{1/2}$$

$$= n^{1+1/2}$$

$$= n^{1.5}$$

T. Complexity = $O(n^{1.5})$

③

```

int i = 1
while (i < n)
{
    i *= 3
}

```

⇒

```

for (i = 1; i < n; i = i * 3)
{
}

```

Time = $O(\log_3 n)$
Complexity

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:-

$Fib(0) = 0$ $Fib(1) = 1$

* $Fib(n) = Fib(n-1) + Fib(n-2)$

$n = 5?$

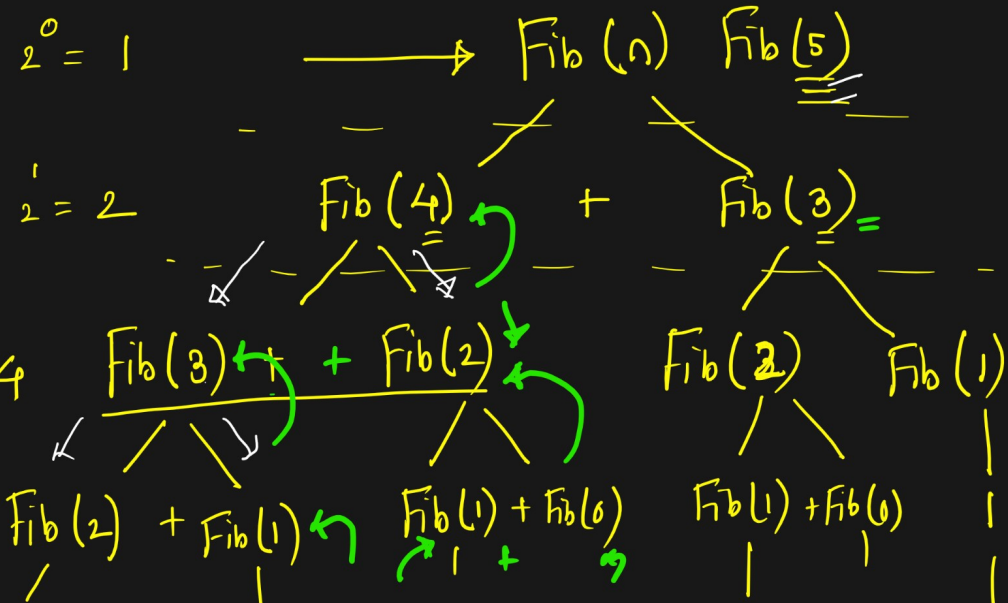
Fibonacci Numbers

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$



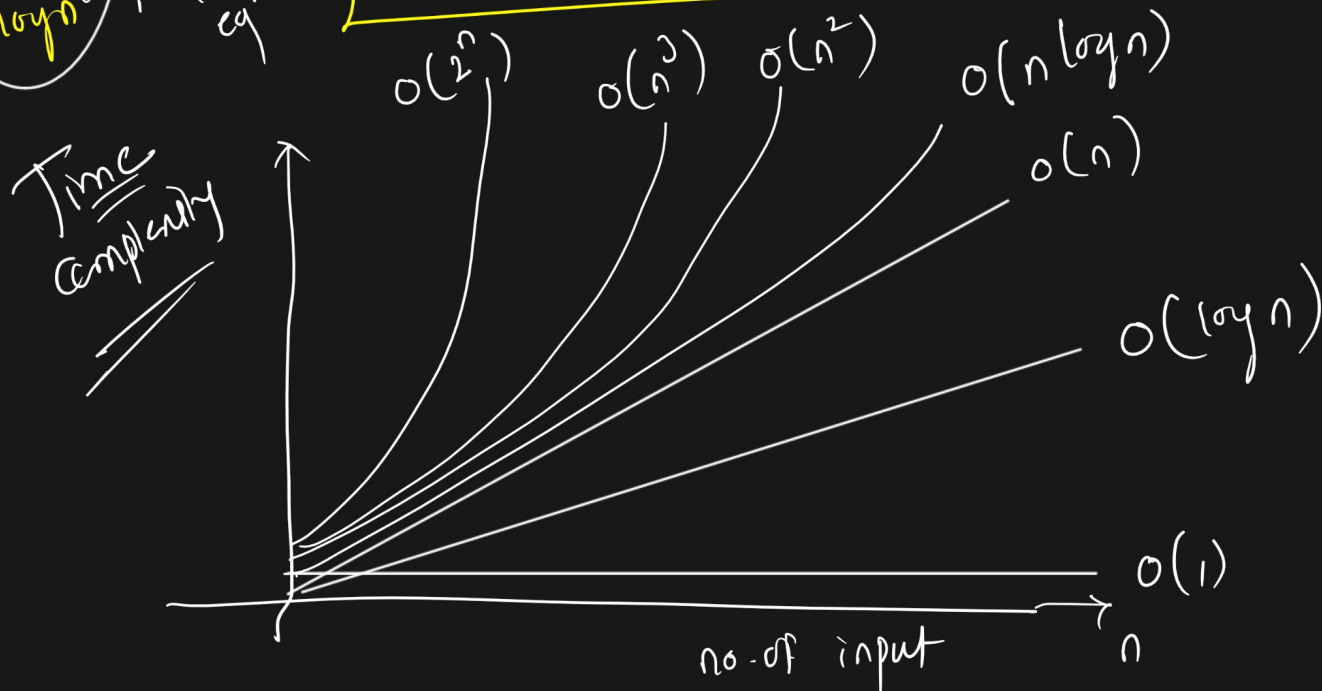
$$\begin{array}{ccccccc}
 4 & & 1 & & 1 & & 1 \\
 2 = 16 & \text{Fib}(1) + \text{Fib}(0) & 1 & & 0 & & 1 \\
 & | & & & & & \\
 5 & \text{Fib}(1) + 0 + 1 & & 1 + 0 & & 1 + 0 + 1 \\
 2 = 20 & & & & & &
 \end{array}$$

0, 1, 1, 2, 3, 5, 8, 13, ...

Time Complexity = $O(2^n)$ = Exponential Eqⁿ

n ✓
 n^2 ✓
 n^3 ✓
 $\log n$ ✓
 $n \log n$ ✓
 2^n → exponential eqⁿ
 polynomial eqⁿ

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$$



Fib(5)

2⁰

2¹

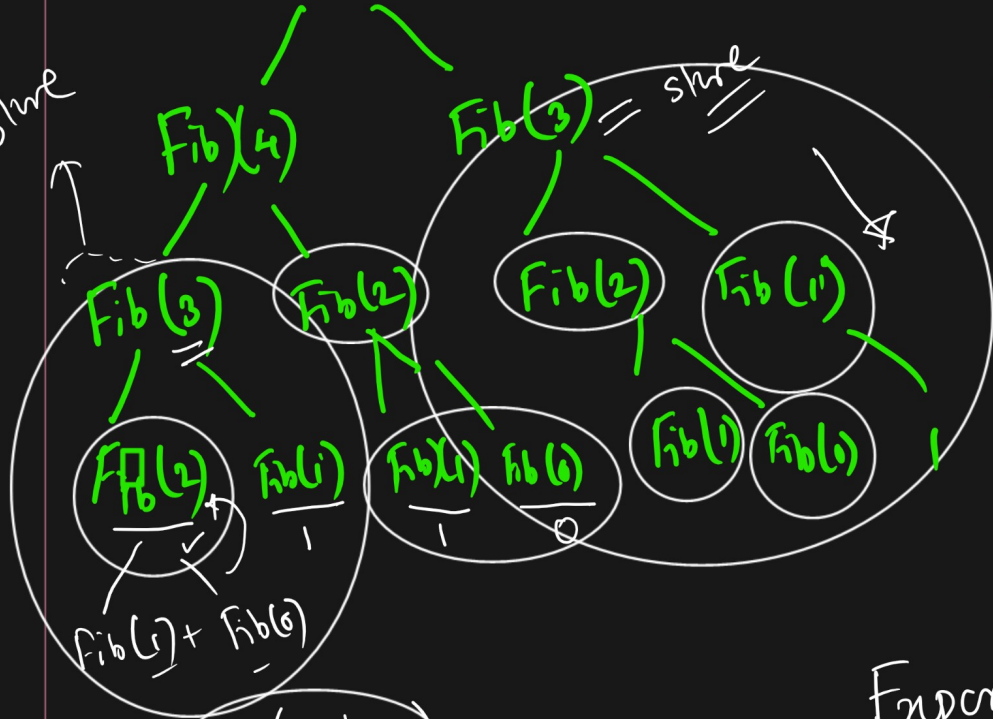
2²

2⁴

When n is high
then it goes to very

$\approx O(2^n) \Rightarrow \text{High}$

How we can reduce



```
int fib(int n)
{
    if (n == 0)
    {
        return 0;
    }
    else if (n == 1)
        return 1;
    [ else
        return fib(n-1) + fib(n-2)
    ]
}
```

Exponential
 $O(2^n)$

Dynamic
Prog ✓

Polynomial
 $= O(n^2 / \log n)$

Dynamic
Prog

[Tabulation
Memoization

② Leetcode Prob No: 231:- Power of Two

$$n \Rightarrow 2^x = n$$

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

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

$$n = 4/2 = \frac{4}{2} = \text{remainder } 0 \\ = 2/2 = 1 \rightarrow \text{True}$$

$$n = \frac{6}{2} = \frac{3}{2} = 1.5 = 0.75 \quad \text{False}$$

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

$$n = 9 = \frac{9}{2} = 4.5 = \frac{2.25}{2} = \frac{1.25}{2} = 0.625 \quad \text{False}$$

16
isPowerOfTwo (int n)

Base
Condⁿ

```

{
    if (n > 1) ✓
        return true;
    else if (n < 1)
        return false;
}
    
```

→ else return isPowerOfTwo ($\frac{n}{2.0}$) ✓

$$\frac{4}{2.0} = \frac{2.0}{2} = \frac{2}{2} = 1$$

$$\frac{16}{2.0} = \frac{8.0}{2} = \frac{4}{2} = \frac{2}{2} = 1$$

int c = 0;

while (n >= 5)

{

n = n / 5

c = c + n;

return c;

n = 10

2 * 5 = 10

while (10 >= 5)

{

n = 10 / 5 = 2

c = 0 + 2 = 2

}

return c;

n = 25

while (25 >= 5)

{

n = n / 5 = 25 / 5 = 5

c = c + n = 0 + 5 = 5

}

return c;

1 * 5 = 5

while (5 >= 5)

{

n = 5 / 5 = 1

c = c + n = 5 + 1 = 6

}

return c;

25 / 5 = 5 + 1 = 6
c = 6

H.W. → Leetcode → 3304