tipe data Var = Val

(1) init Func
(2) Base case
(3) Rekurens

Big-O Notation ⇒ TLE

O(...)

↳ kompleksitas

↳ $10^8 \approx$ 1 detik

$> 10^8 >$ 1 detik

(1) Jumlah iterasi

- ↳ Loop
- ↳ Recursive

(1) Konstan ⇒ O(1)

if else
deklarasi variable
cin cout
assignment

(2) Linear ⇒ O(N), O(2N), O(M+N), ...

```
for(int i = 1; i<=N; i++){
cout<<"Gas"<<endl;
}
```

} N-iterasi ⇒ O(N)

```
for(int i = 1; i<=N; i++){
   for(int j = 1; j<= M; j++){
     cout<<"Gas"<<endl;
   }
}
```

⇒ O(NM)

(3) Polinomial

```
for(int i = 1; i<=N; i++){
  for(int j = 1; j<= i; j++){
    cout<<"Gas"<<endl;
  }
}
```

i = 1 j = 1 → gas

i = 2 j = 1 → gas gas
j = 2

i = 3 j = 1
j = 2
j = 3
} gas gas gas

Setiap i akan dicerak "gas" sebanyak i kali ⇒ $1 + 2 + 3 + \ldots + N$

$$\text{Jml iterasi} = \frac{n * (n-1)}{2} = \frac{n^2 - n}{2}$$

$O(N^2)$

(4) Logaritmik

N akan di'bagi' 2 terus = 0

```
int N;   ↩ stop = 0
while(N/=2){
  cout<<"gusdim"<<endl;
}
```

iterasi N → 1

sebanyak x

$$N / 2 / 2 / 2 / 2 / 2 \dots / 2 = 1 = N / 2^x$$

$$N * 2 * 2 * 2 * \dots * 2 = N * 2^x$$

sebanyak x

$$\frac{N}{2^x} = 1 \rightarrow N = 2^x$$

x = jml iterasi

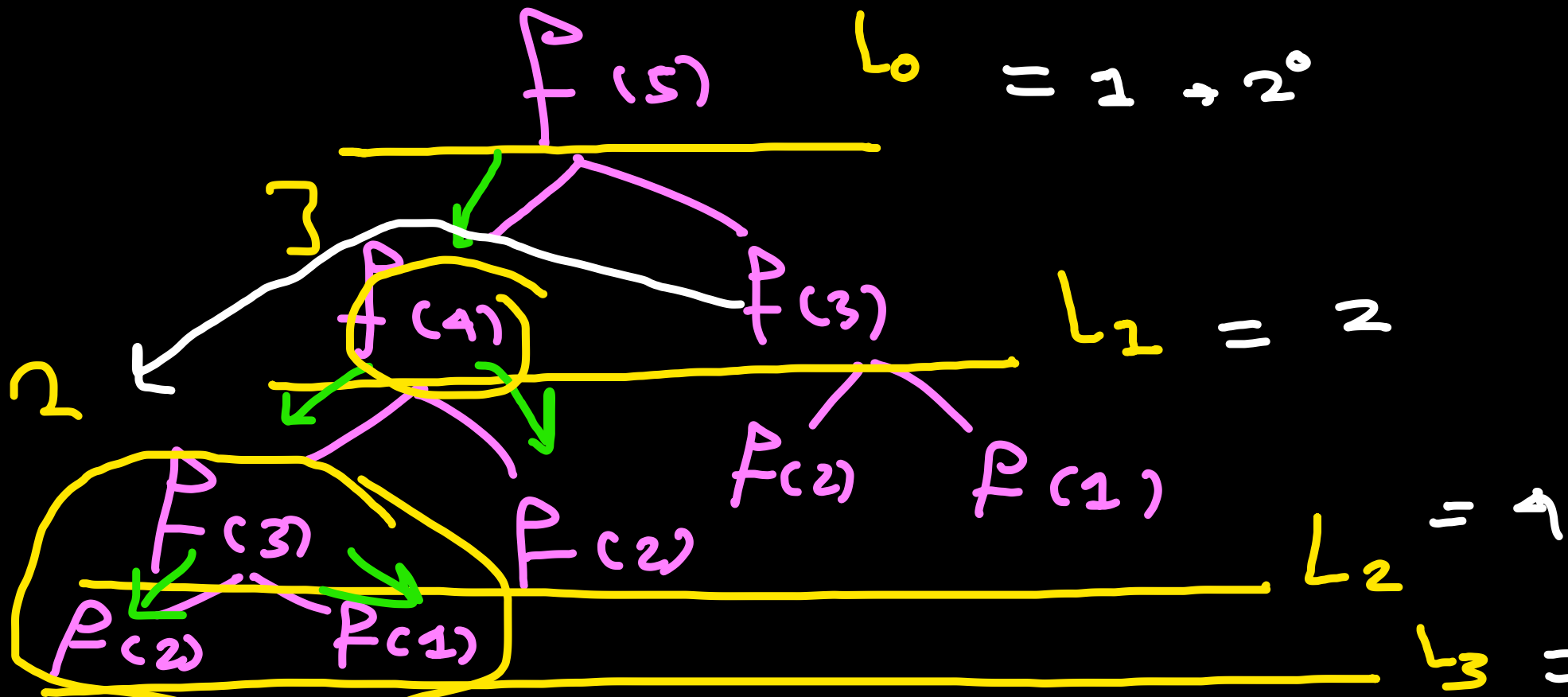
$2^x = N \longrightarrow x = 2^{\lg N}$

$x = \lg N$

$O(\lg N)$

(5) Exponential

$$f(n) = f(n-1) + f(n-2)$$



Per level $\Rightarrow 2^N \Rightarrow O(2^N)$

MEMO[n] = save fact(n)

TC: 6

1 ⇒ Fakt(1) = 1

2 ⇒ Fakt(2) → F(1) → 1 = 2*1

3 ⇒ Fakt(3) → Fakt(2) → Fakt(1) → 1 = 6

4 ⇒ Fakt(4) → Fakt(3) ⇒ Fakt(2) ⇒ Fakt(1) ⇒ 1 = 24

4*6

5 ⇒ Fakt(5) ⇒ Fakt(4) ⇒ ... ⇒ 1

Fakt(3)

6 ⇒ Fakt(6) ⇒ Fakt(5) ⇒ ... ⇒ 1

5*24=120

6 * 120 = 720

MEMO
1 → 1
2 → 2
3 ⇒ 6