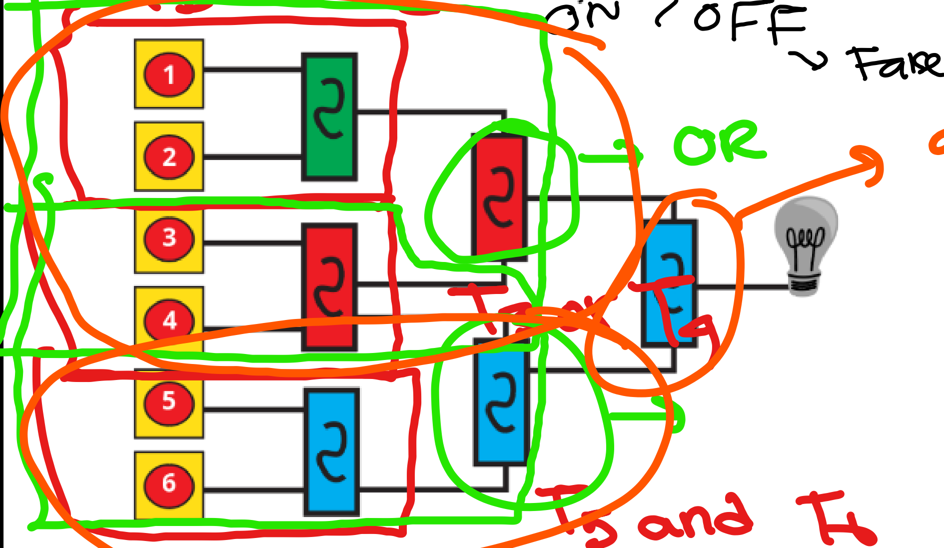
*Deskripsi soal berikut untuk nomor 1 – 3!*

**Tekan – Tekan Saja**




Pak Dengklek mempunyai rangkaian listrik yang terdiri dari enam buah tombol dan jika ditekan akan mengalirkan arus listrik pada kabel – kabel terhubung. Namun arus yang mengalir akan melewati gerbang controller dengan 3 jenis yaitu : Gerbang yang bisa dilewati satu buah arus terhubung saja (hijau), Gerbang yang bisa dilewati minimal satu arus terhubung (merah), dan gerbang yang hanya bisa dilewati jika arus yang terhubung adalah dua arus sekaligus (biru)

1. Dari permasalahan di atas penekanan tombol 1 – 2 – 3 – 4 – 5 – 6 secara berurutan yang benar sehingga lampu dapat menyala adalah ...
   a. ON – OFF – OFF – ON – ON – ON
   b. ON – OFF – OFF – OFF – ON – ON
   c. ON – ON – OFF – ON – OFF – ON
   d. ON – ON – ON – OFF – ON – OFF
   e. OFF – ON – OFF – OFF – OFF – OFF

**JAWABAN A**

Minimal satu = OR

harus satu = XOR

gaboleh False

$$\Big(\underbrace{(T_1 \text{ xor } T_2) \text{ or } (T_3 \text{ or } T_4)}\Big) \text{ and } \Big(\big(\underbrace{\overset{F}{T_3} \text{ or } \overset{F}{T_4}}\big) \text{ and } \big(\underbrace{T_5 \text{ and } T_6}\big)\Big)$$

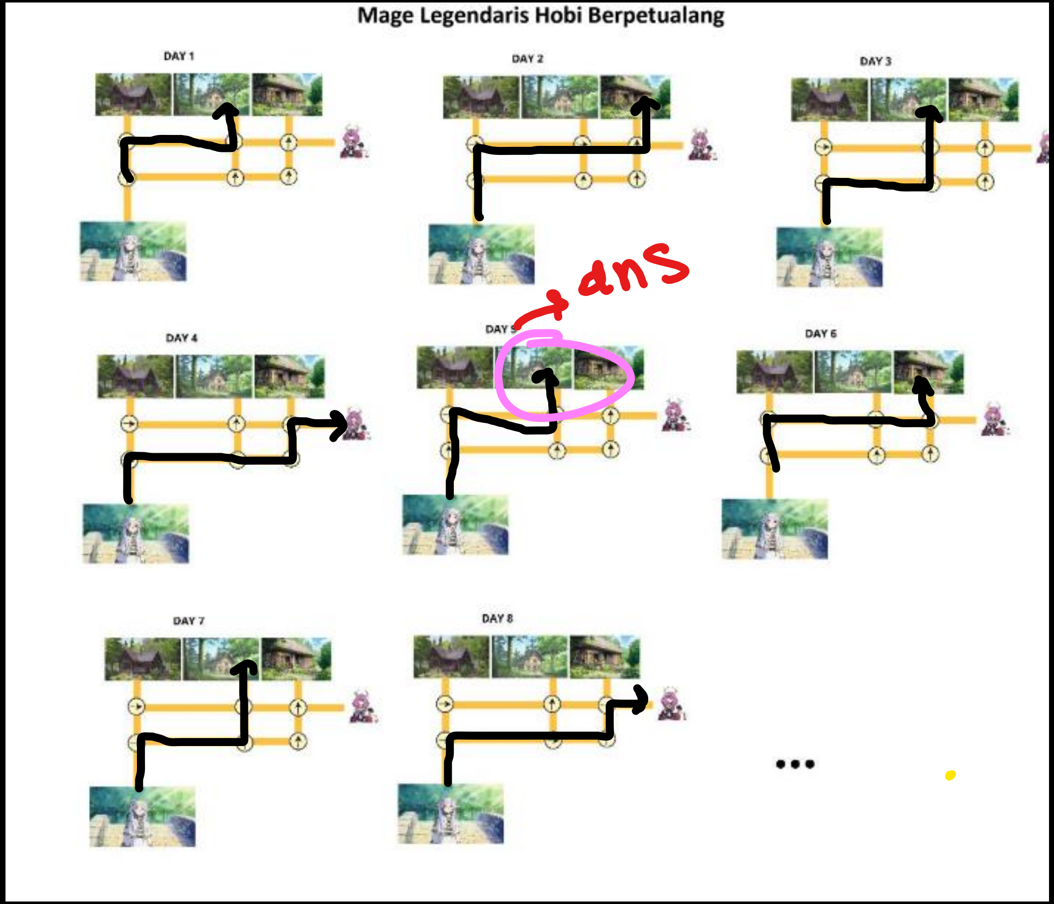
gaboleh

gaboleh

False

$T_5 = F$ atau $T_6 = F$

$T_5$ dan $T_6$ = True




4. Berdasarkan gambar di atas jika Frieren berjalan pada hari $5^{2023} + 24^{25^{26}}$ Frieren akan berada di ...
   a. Rumah Fern
   b. Rumah Stark
   c. Rumahnya Sendiri
   d. Bertemu Demon Jahat
   e. Tidak dapat ditentukan

$(5 \bmod 8) \cdot \underline{(5^{2022} \bmod 8)} \bmod 8$

$(5^2)^{1011} \bmod 8$

$25^{1011} \bmod 8 \equiv (25 \bmod 8)^{1011}$

$\equiv \underline{1}$

hari mod 8

$5^{2023} + 24^{25^{26}} \bmod 8$

$\left(\left(5^{2023} \bmod 8\right) + \left(\cancel{24^{25^{26}} \bmod 8}\right)\right) \bmod 8 = 5^{2023} \bmod 8$

$(5 \cdot \cancel{5^{2022}}\ 1 \bmod 8) = 5$

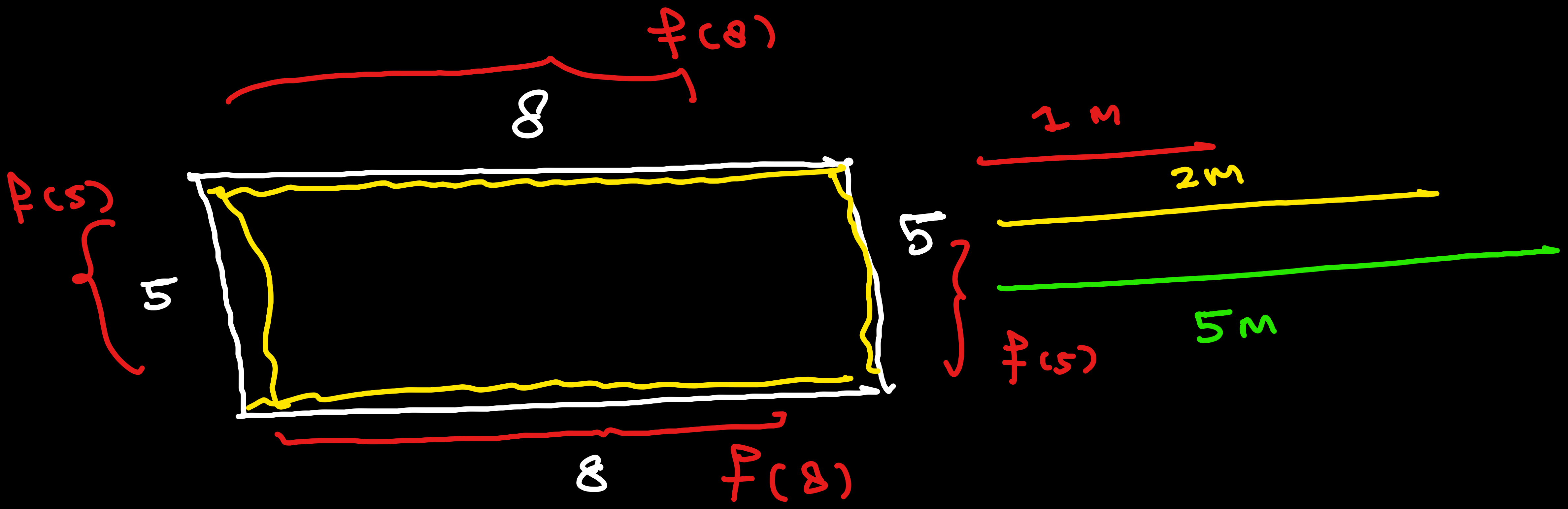
$$a \overset{\times}{\underset{-}{+}} b \bmod n \equiv ((a \bmod n) \overset{\times}{\pm} (b \bmod n)) \bmod n$$

$$99999 \bmod 5 = ((9\cancel{9990 \bmod 5}) + (9 \bmod 5)) \bmod 5$$

$$= 9 \bmod 5$$

$$= \underset{\sim}{4}$$

$$a^b \bmod n \equiv (a \bmod n)^b \bmod n$$




* Dynamic Programming ⇒ Recursive

$$f(8) + f(8) + f(5) + f(5)$$
$$= f(8)^2 + f(5)^2$$

1m

2m

5m

$n$

$f(n)$ = banyak cara memasang pagar pada perimeter berukuran $n$

1m $\underbrace{\qquad\qquad}_{n-1}$ $\rightarrow f(n-1)$

2m $\underbrace{\qquad\qquad}_{n-2}$ $\rightarrow f(n-2)$

5m $\underbrace{\qquad\qquad}_{n-5}$ $\rightarrow f(n-5)$

$$f(n) = \underline{f(n-1)} + \underline{f(n-2)} + f(n-5)$$

$f(0) = 1$

$f(1) = 1$, $f(2) = 2$

$f(3) = 3$, $f(4) = 5$

1 1
2

1 1
2
} 2

1 semua
1 1 2
2 1 1
1 2 1
2 2
} 5 cara

```
int base_case[5] = {1,1,2,3,5}

if(n<5){
  return base_case[n];
else{
  return ...
```

$f(8) = \ldots$

$f(5) = \ldots$

# Kombinatorika

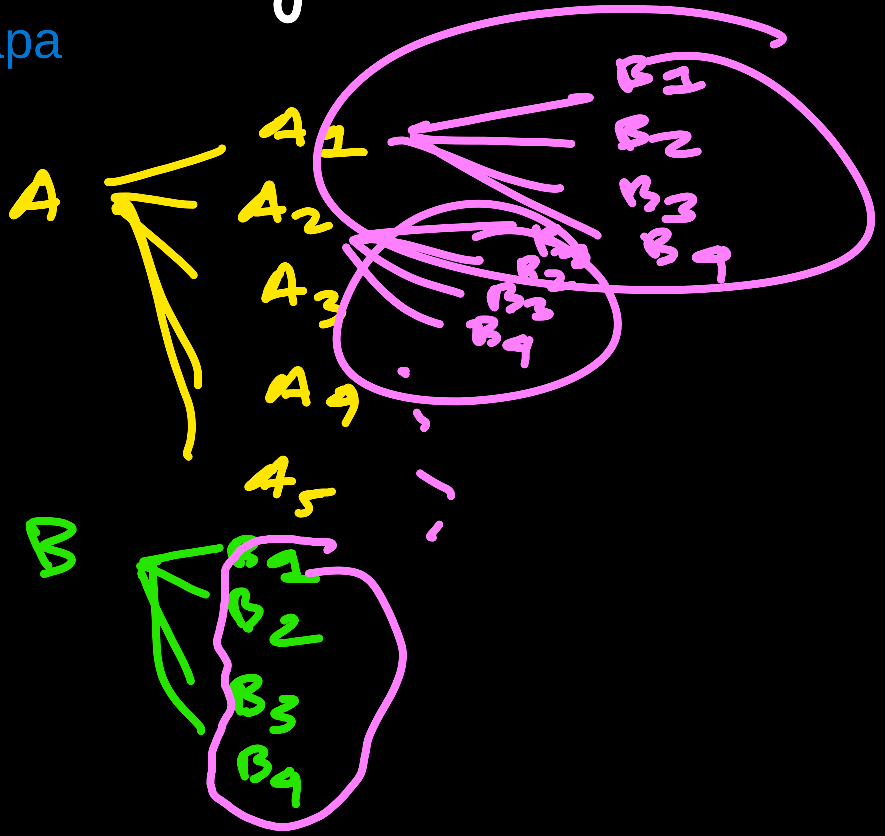
* Aturan Perkalian

$K_1 \longleftrightarrow K_2$ (saling berkaitan)

Punya irisan

Ada 5 jalan dari A ke B, 4 jalan dari B ke C, berapa banyak jalan dari A ke C tanpa putar balik

$5 \times 4$



# Analisis Kompleksitas

Diberikan input bilangan bulat N
hitung berapa
$1 + 2 + 3 + 4 + .. + N$

gauss summation

Cara 1 :

```
for(int i = 1; i<=N; i++)
    sum+=i
```

Cara 2 :

```
cout<<n*(n+1)/2<<endl;
```

$O(\ldots) =$

kompleksitas

Big-O Notation $O(\ldots)$

$10^{8} \Rightarrow 1$ detik

* Konstan → $O(1)$

define variable, array, tipe data, dll, operasi (TIDAK ADA LOOP, TIDAK ADA RECURSIVE)

int n = ...

if(...)

else

cout<< ... <<<

* Linear - Polinomial

10 iterations

```
for(int i=1; i<=10; i++){
cout<<"Roblox"<<endl;
}
```

iteration

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

O(N)

→ N orang , Setiap orang sebut 3x Roblox

```
for(int i = 1; i<=N; i++){
 for(int j = 1; j<=3;j++){
    cout<<"Roblox"<<endl;
 }
}
```

$O(3N)$

$10^4 \leq 10^8$

1s

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

$O(N^2)$

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

$O(NM)$

worst ← $i_{max} = N$,

$j_{max} = i_{max}$

$j_{max} = N$

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

$O(N^2)$

$$1 + 2 + 3 + \dots + N = \frac{N*(N+1)}{2}$$

$$= \frac{N^2 + N}{2}$$

$1 <= N <= 10^8$ $\qquad O(N^2)$

1 detik $\qquad O(\ldots)$

$\hookrightarrow \le 10^8$

$O((10^8)^2) = O(10^{16})$ $\boxed{\text{TLE}}$

$\hookrightarrow$ > 1 detik

$1 \le N \le 10^9$, $O(N^2)$

$10^{18} > 10^8 > 1s$

$f(1) = 1, \ f(2) = 1$

## * Eksponensial

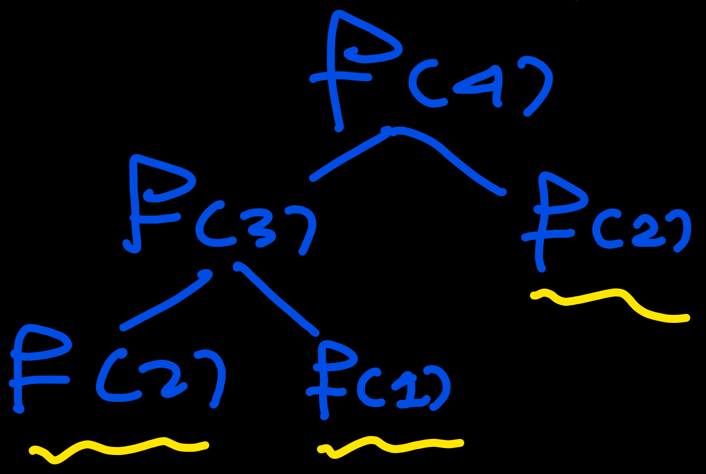
```
int f (int n){
  if(n == 1 || n == 2) return 1;
  return f(n - 1) + f(n - 2);
}
```

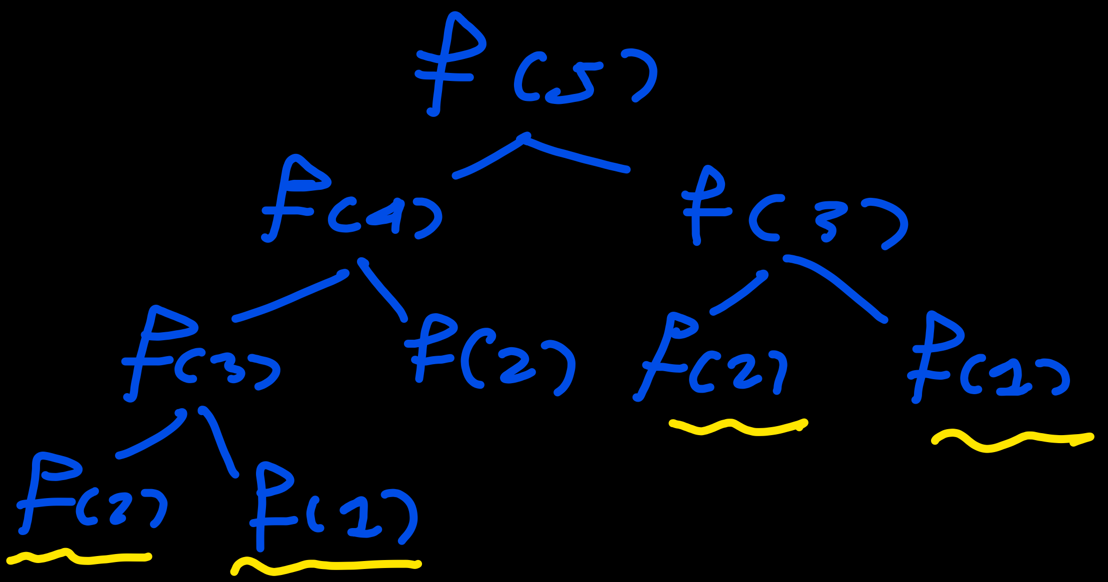
Base case f(1) dan f(2)

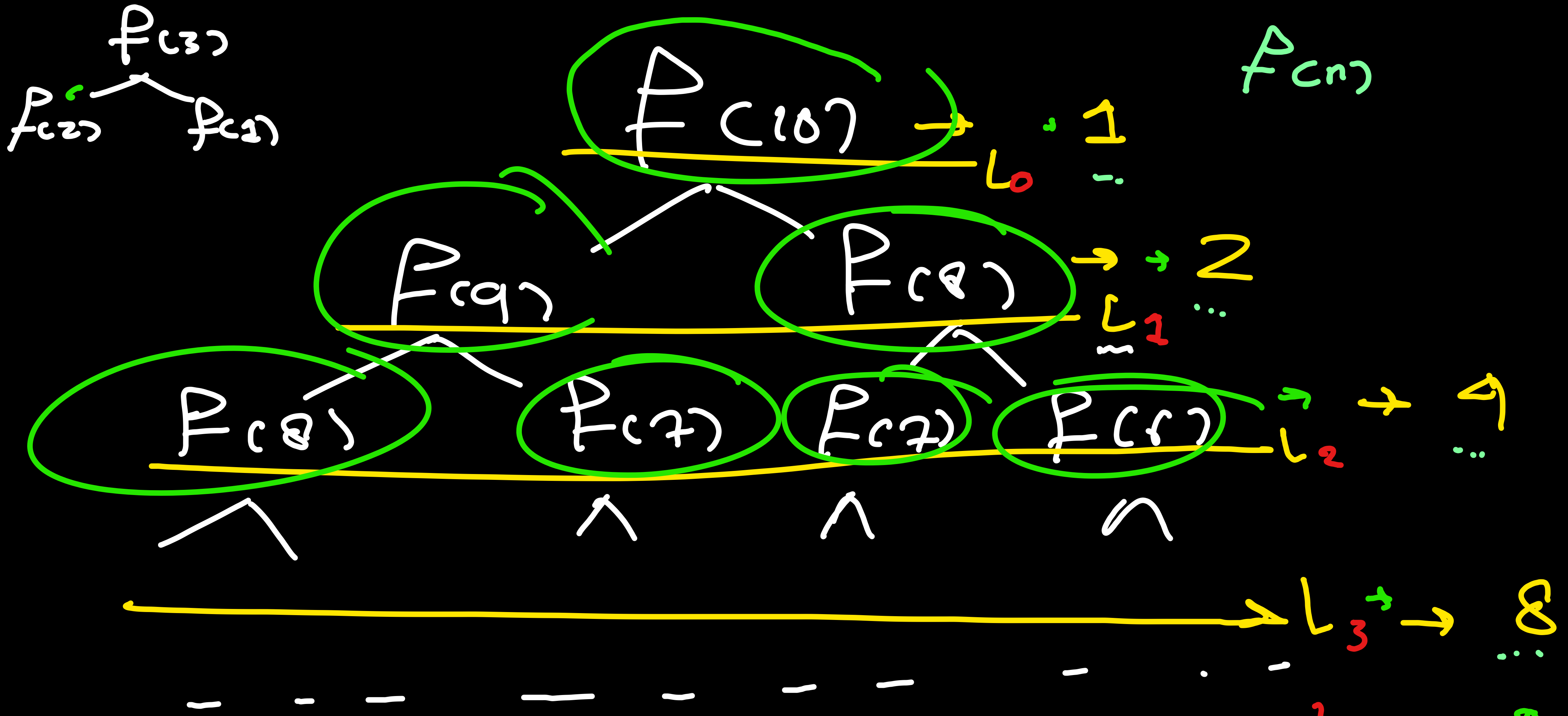
Rekurens

Fibonacci

Berapa kali manggil function






F(1) F(2) $O(2^n)$ $L_9 \to 2^9$

$10 \to 2^9$

$n \to 2^n$

di Level ke-n Manggil function $2^n$

$O(2^N)$

* Logaritmik

```
int N;
cin>>N;
while(N/=2){
  cout<<"Axel""<endl;
}
```

Jalan ketika = true

!= 0

while ( ... ) {

}

Stop ketika = False

= 0

axel akan dicetak terus terakhir

N = 1

$$N \underbrace{* 2 * 2 * 2 * 2 \ldots *}_{\text{sebanyak } x} 2 = N \cdot 2^x$$

iteration

$$N \underbrace{/2 /2 /2 /2 /2 \ldots}_{\text{sebanyak } x} /2 = \frac{N}{2^x}$$

brp kali Arr dicotok

$x$ = banyak iteration

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

$$x = {}^{2}\log N \rightarrow O(\lg N)$$

$$O(\log N)$$

$$a^x = N \leftrightarrow \underline{x = a^{\log N}}$$

```
int T(int N){
 if(N == 1){
     return 1;
 }else{
    return T(N/2) + N;
 }
}
```

$\} \; O(\lg N)\;?$

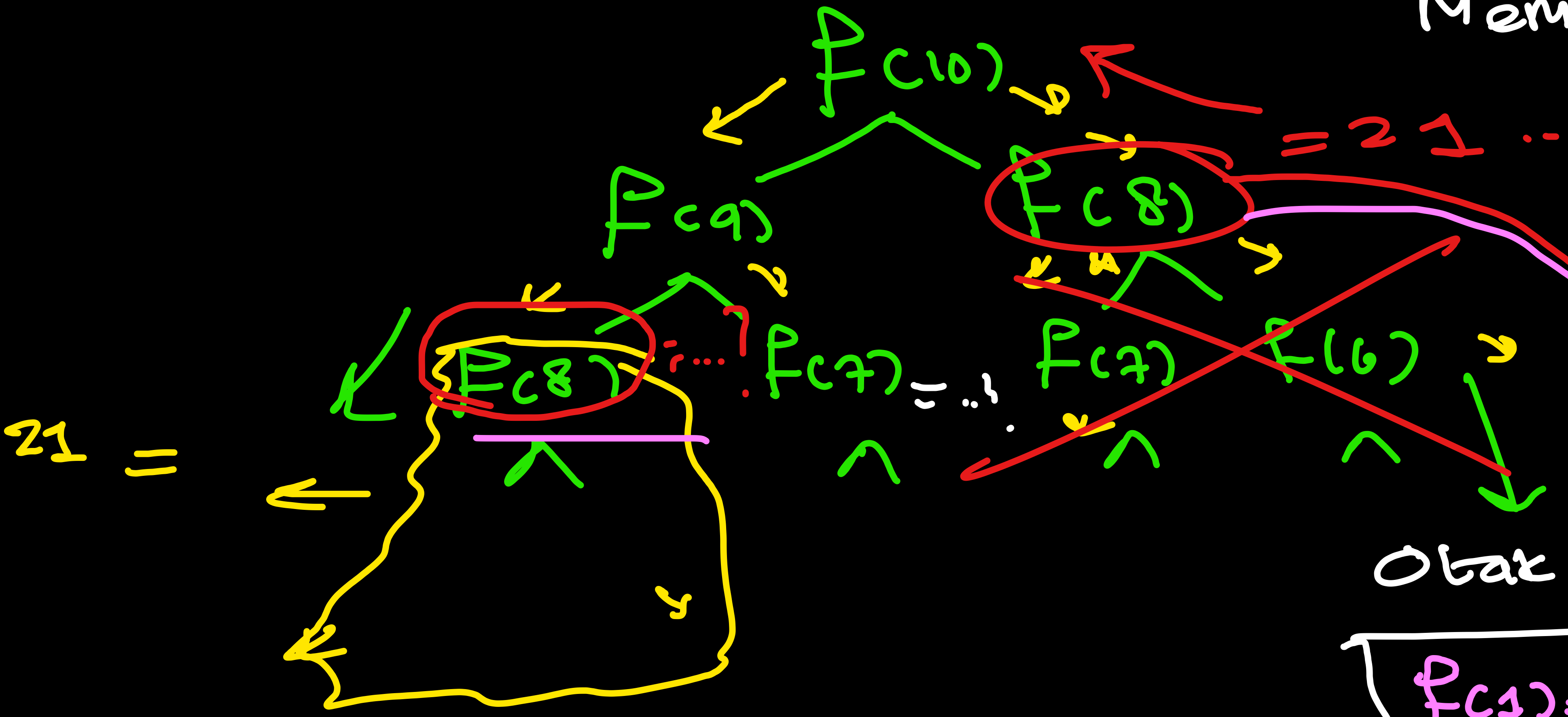
# Dynamic Programming

$$f(n) = f(n-1) + f(n-2), \quad f(1) = 1, \; f(2) = 1$$

kompleksitas biasa $\Rightarrow O(2^n)$

1, 1, 2, 3, 5, 8, 13, 21

Memo ...




Otak / Memo

$f(1) = 1$

$f(2) = 1$

$\vdots$

$f(8) = 21$

F(10)

↓

F(9)

↓

F(8)

↓

F(7)

↓

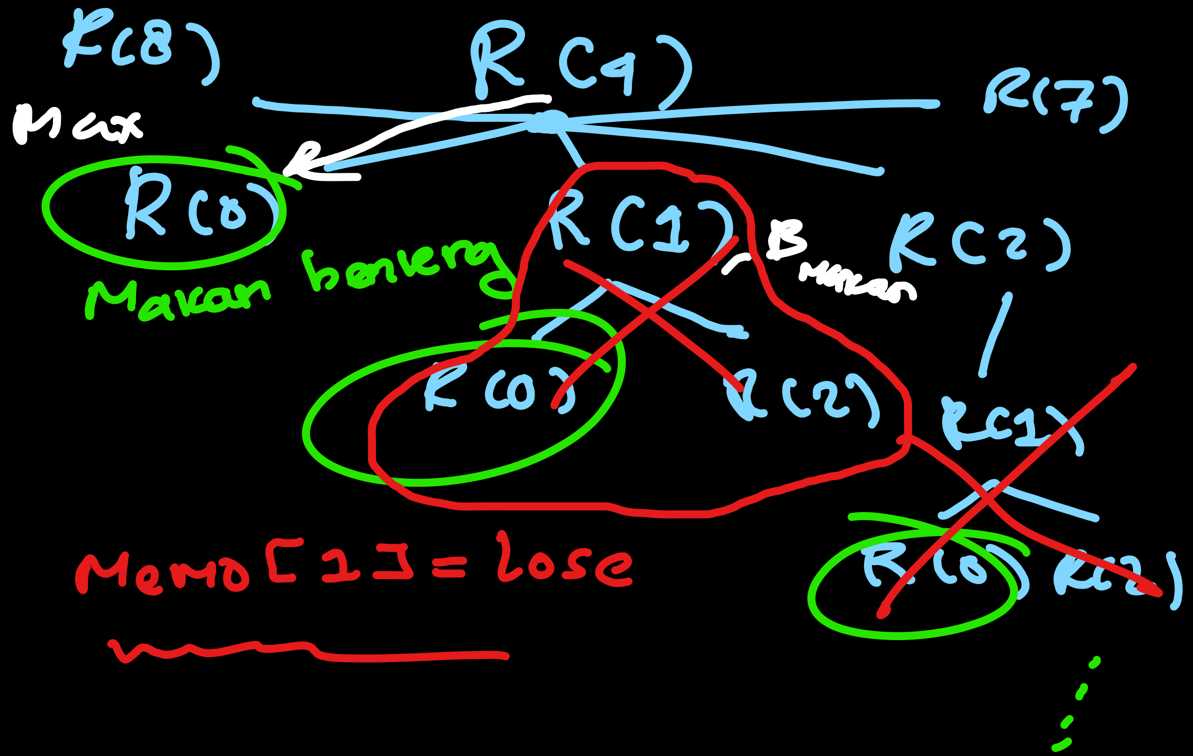
F(6)

↓

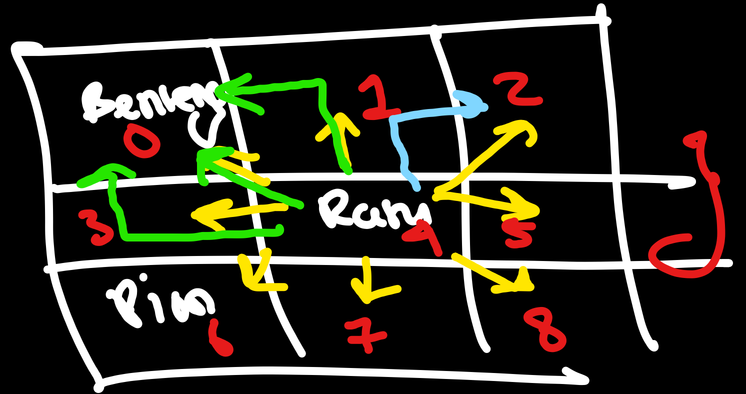
⋮

F(1)

O(N)




Memo[1] = lose




8 langkah

Soal Rekursif → Dynamic Programming
→ DnC, Greedy, etc

(1) opitimausasi → min, max

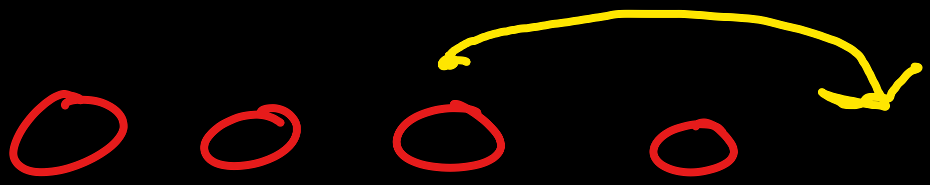
(2) Traverse → movement
→ search

(3) Combinarorics → Berapa banyak cara
→ Berapa banyak susunan

$$f(n) = n * f(n-1), \quad f(1) = 1$$

$$C(n,r) = \frac{n!}{(n-r)! \cdot (r!)}$$

$$= \frac{f(n)}{f(n-r-1) \cdot f(r)}$$



out of bound

$B_1 \to B_2$

$B_1 \to B_3$

$B_1 \to B_2$ (Loncat 1 Petak) $\Rightarrow B_i \to B_{i+1}$

$B_1 \to B_3$ (Loncat 2 Petak) $\Rightarrow B_i \to B_{i+2}$

N batu

10 30 40 20


* cost minimum → $\min(B_1 \to B_2 \to B_3,\ B_1 \to B_3)$

$B_1 \to B_2 = B_1 \to B_2 = |30-10| = 20$

$B_1 \to B_3 =$ 1 langkah $B_1 \to B_2 \to B_3 = |30-10| + |40-30| = 30$

2 langkah $B_1 \to B_3 = |40-10| = 30$

* Model Matematis → Fungsi Rekursif

(1) Bikin Fungsinya ⇒ $f(n)$

(2) Tentukan Base case

(3) Rekursi / Rekurens

$f(n)$, $B_1 - B_n$

$B_1 - B_n \Rightarrow f(n)$

$B_1 \to B_2 = 30$

(1) $f(n)$ = Menyatakan berapa cost min dari $B_1 \to B_n$

(2) $f(1) = 0$

$f(2) = 20$

$f(3) = 30$

$f(4) = 30$

$f(n) = B_1 \to B_n$

$f(2) = B_1 \to B_2 = 20$

$B_1 \to B_3 = 20$

$f(3) = 20$

$\text{cost}(B_i \to B_j) = |h_i - h_j|$

$f(n) = B_1 \to B_n$ $\quad f(1) = B_1 \to B_1$

$f(1) = B_1 \to B_1 = |h_1 - h_1| = 0$

$f(1) = 0$

(1) Top-down

(2) Bottom-up

(3)

(1) Teknik pola

$f(1) = 1$
$f(2) = 1$
$f(3) = 2$
$f(4) = 3$

$2 = 1 + 1$
$f(3) = f(2) + f(1) \rightarrow f(n) = f(n-1) + f(n-2)$
$f(4) = 2 + 1$
$f(4) = f(3) + f(2)$

(2) Teknik intuitif – deduksi induksi

$f(n+1)$

$f(n) = \min(\text{cost}(\text{loncat 1 langkah}),$
$\text{cost}(\text{lonjat 2 langkah}))$

$+ |\text{cost}[x][y]|$

$f(n+2)$

f(x,y)

$$f(n) = B_1 \to B_n$$

$$f(x,y) = B_x \to B_y$$

$$f(1,1) = 0$$

$$f(1,2) = 20$$

$$f(1,3) = 30$$

## Transisi state

$$f(x,y) = \min(\text{Loncat 1 langkah}, \text{loncat 2kali})$$

$$=$$

$$B_x \to B_{x+1} = f(x+1, y)$$

$$B_x \to B_{x+2} = f(x+2, y)$$

start - finish

bottom up

$$f(x,y) = \min\left( f(x+1, y),\ f(x+2, y) \right) + \mathrm{cost}(x, y)$$

$$f(x,y) = \min\left( f(x+1, y),\ f(x+2, y) \right) + |h_x - h_y|$$

bottom ⇒ start = finish return 0

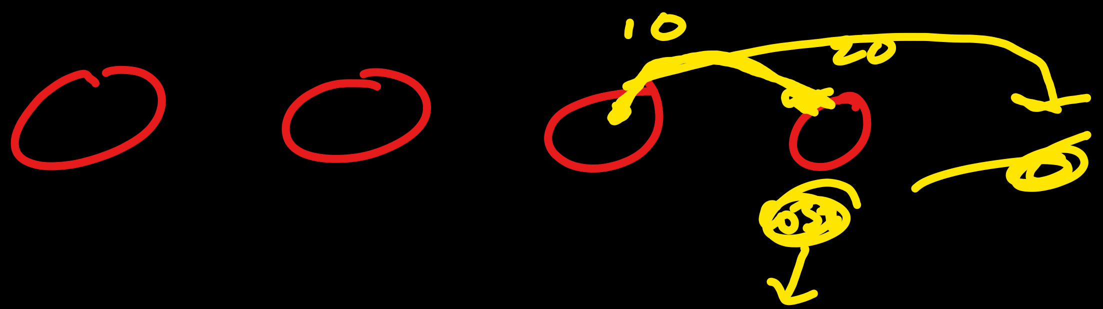
base case = $x = y$

$f(5,5) =$

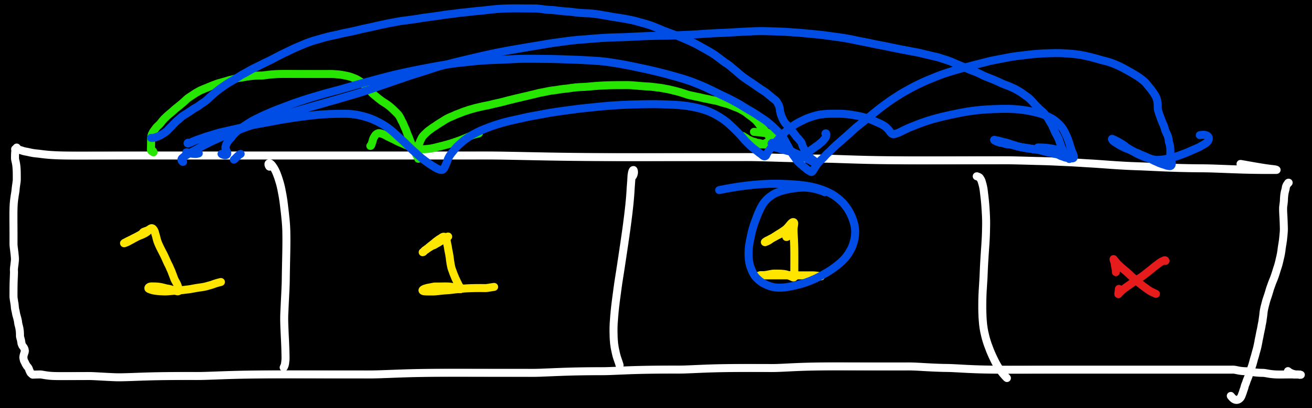
$$f(x,y) = |b_y - b_x|$$

Top down

10 0
min( 2l, 2l ) = 0




20 - 10 = 10




dp[i-2]

State top down

$dp[n]$ = Berapa banyak kombinasi Penjumlahan $(= n)$

$$\underbrace{\_\_ + \_\_ + \_ + \_ + \_\_}_{dp[n]} = n$$

$$1 + \underbrace{\_\_ + \_ + \_ \;\; + \_}_{dp[n-1]} = n - 1$$

$$2 + \underbrace{\_ + \_ \_ + \_ + \_}_{dp[n-2]} + \_$$

$\vdots$

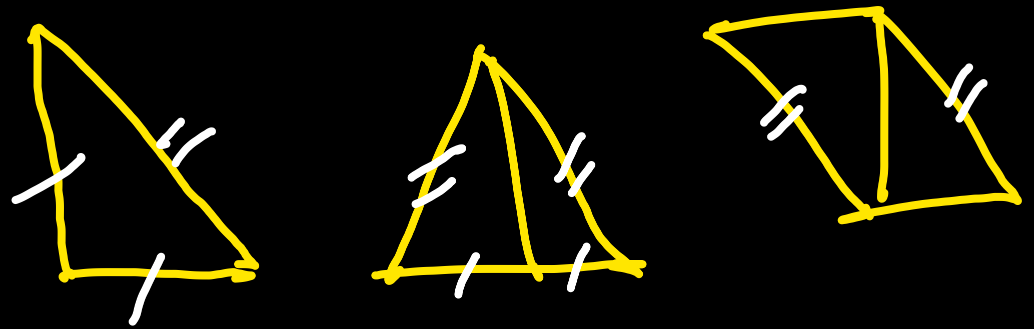
$$6 + \underbrace{\ldots + \ldots + \ldots}_{dp[n-6]} + = n-6$$

$$dp[n] = \sum_{i=1}^{6} dp[n-i] \quad , \quad \begin{array}{l} dp[0] = 0 \\ dp[1] = 1 \\ dp[2] = 1 \end{array}$$

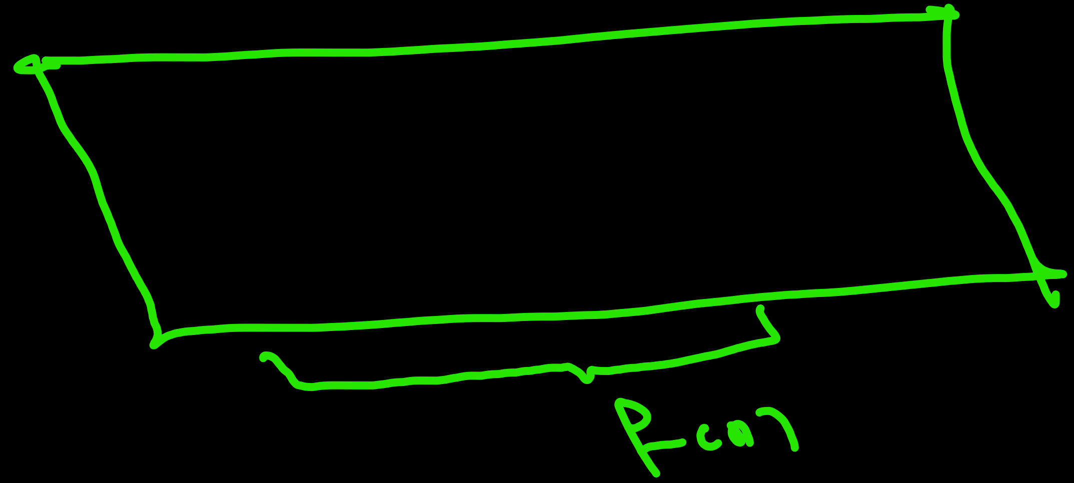
$n - i \leq 0$ , return 0

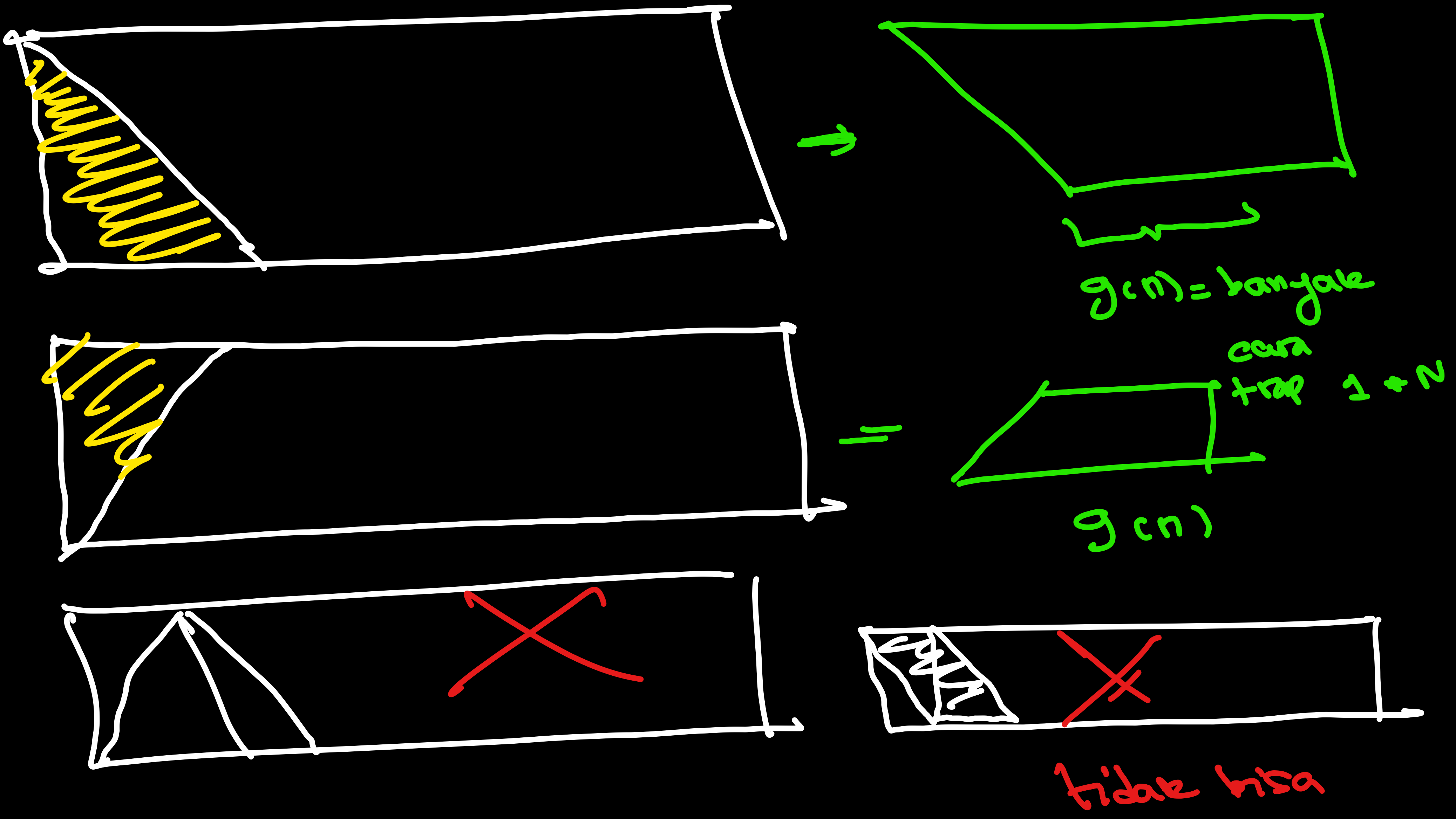
$n = 3$ , $i = 6$

$6 + \cancel{\ldots \leq} 3$ Banyak cara = 0

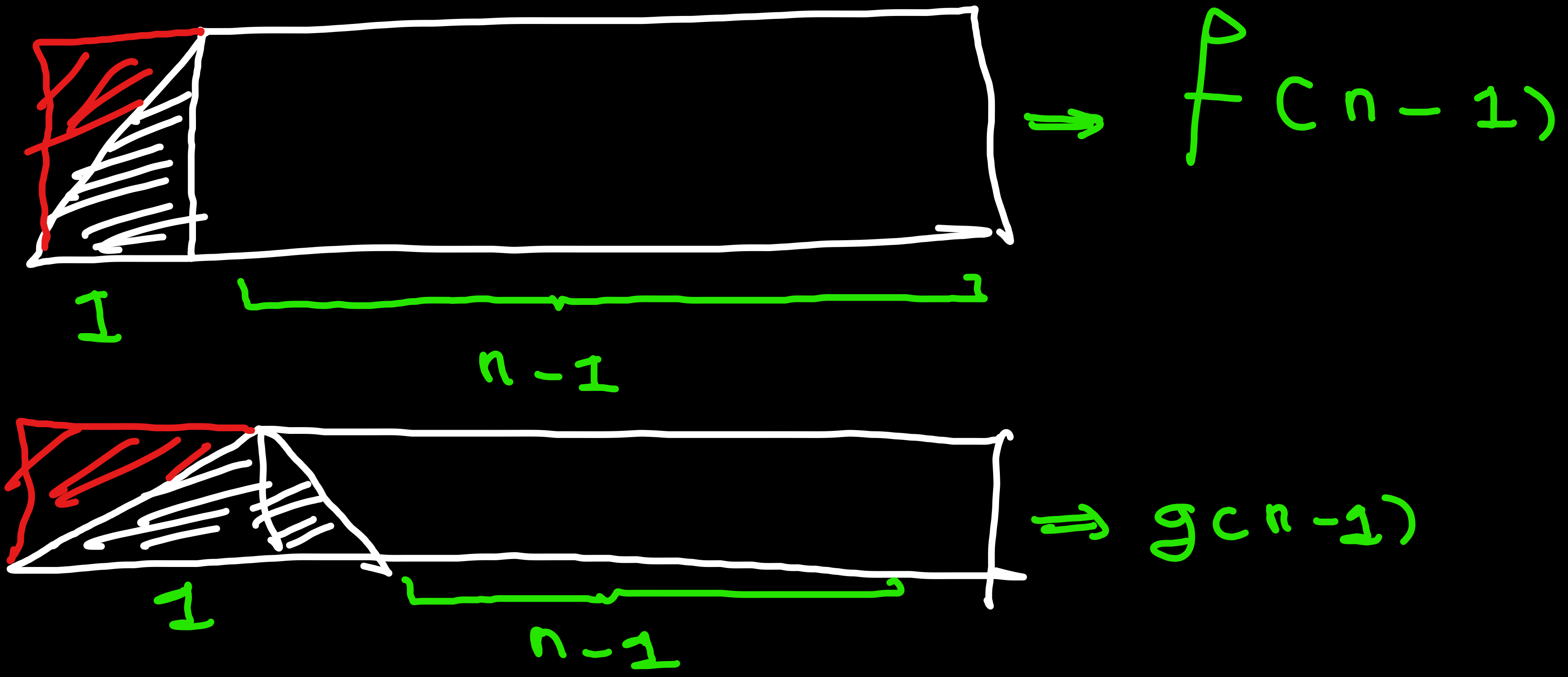



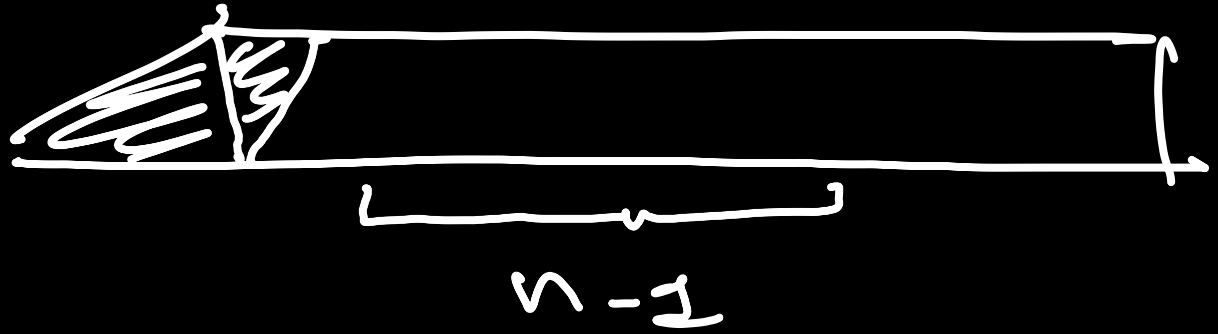
(1) $f(n)$ = Banyak cara memasang ubin pada Lantai $1 \cdot N$ (boleh Rotasi / Refleksi)




g(n) = banyak
cara
trap 1 * N
g(n)
tidak bisa

$$f(n) = 2g(n)$$






$\Rightarrow g(n-1)$

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

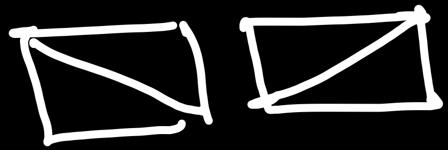
$$f(n) = 2g(n) \Leftrightarrow 2g(n-1) = f(n-1)$$

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

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

$$f(n) = 2 \cdot 2f(n-1) = 4f(n-1) \rightarrow \text{impisit}$$

$$f(1) = 2 = 2^1$$



(2 cara)

$$f(2) = 4 \cdot 2 = 8 = 2^2 \cdot 2^1 = 2^3$$

$$f(3) = 4 \cdot 8 = 32 = 2^2 \cdot 2^3 = 2^5$$

$$f(4) = 4 \cdot 32 = 2^2 \cdot 2^5 = 2^7$$

$$f(n) = 2^{\text{bilangan ganjil ke-}n} = 2^{2n-1} \rightarrow \text{ecspisit}$$