$x \in R$

Floor = $\lfloor x \rfloor$ → membulatkan x ke bawah

Ceil = $\lceil x \rceil$ → membulatkan x ke atas

$\lfloor 3{,}5 \rfloor = 3$

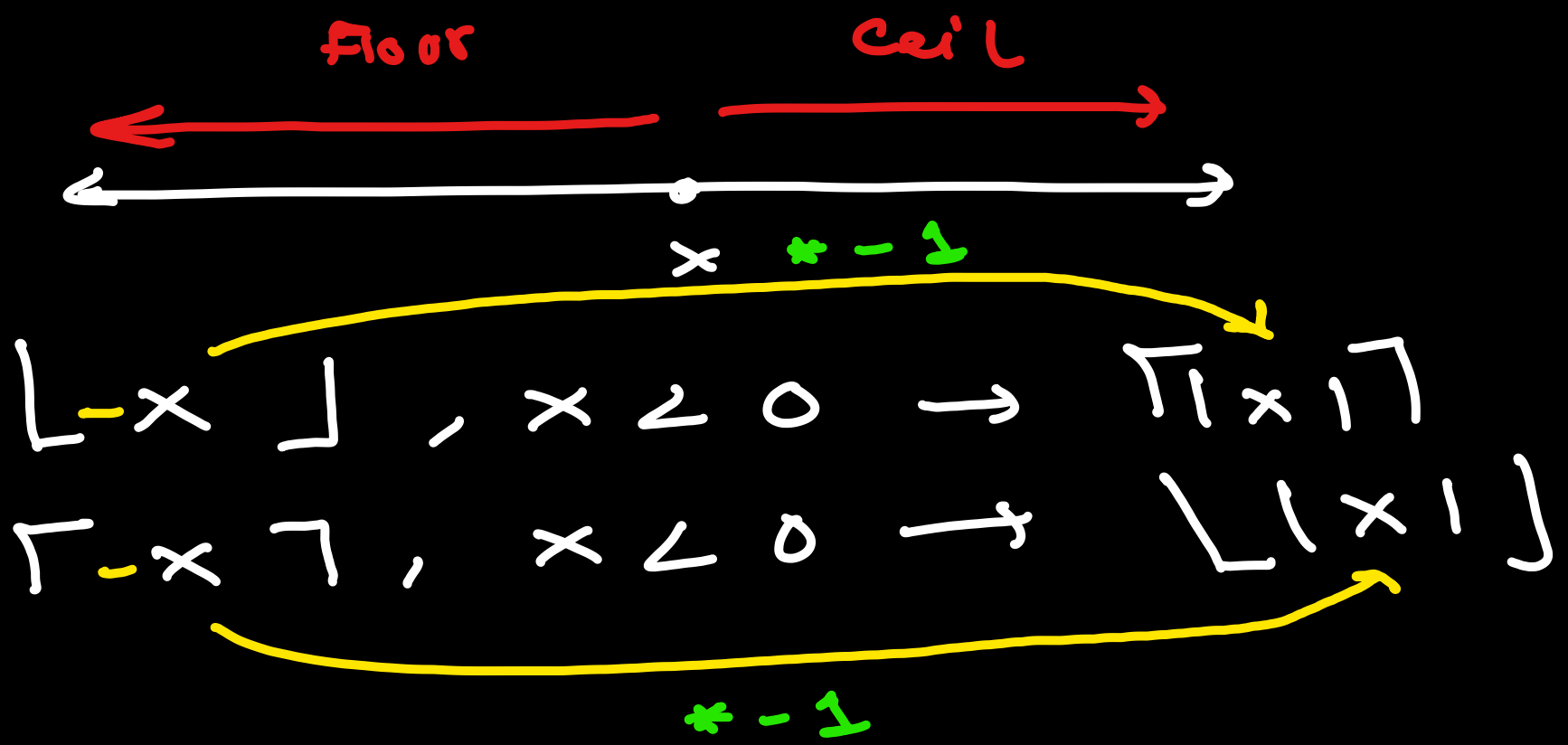
$\lceil 3{,}2 \rceil = 4$

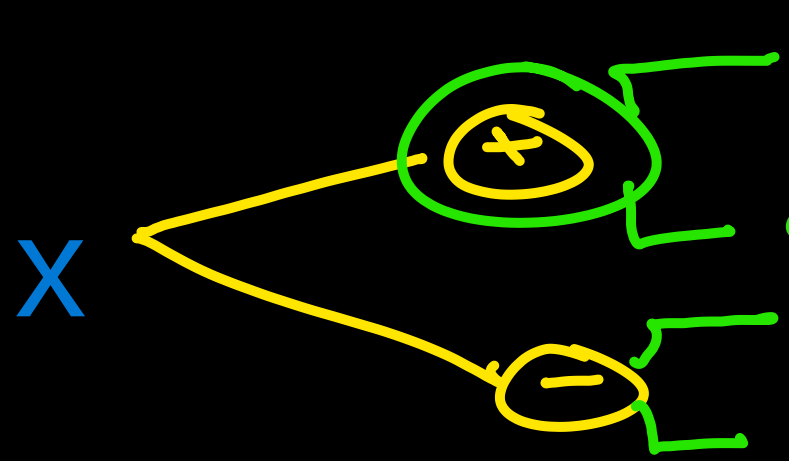
$\lfloor -3{,}5 \rfloor = -4$

$-2 > -3$

$\lceil -2{,}4 \rceil = -2$

$-2 > -2{,}4$


floor
ceil
x * -1
⌊-x⌋ , x < 0 → ⌈|x|⌉
⌈-x⌉ , x < 0 → ⌊|x|⌋
* -1



Floor : ubah tipe data x jadi integer

Ceil : ubah tipe data x jadi integer + 1

Floor : ubah tipe data x jadi integer + 1

Ceil : ubah tipe data x jadi integer

div

int x = a / b

Dipaksa bulat ke bawah

$n = 3{,}5$

$x = \text{trunc}(3{,}5)$

$= 3$

$3{,}5 \neq 3 \rightarrow$ n desimal

$n = 3{,}0$

$x = \text{trunc}(3{,}0)$

$= 3$

$3{,}0 = 3 \rightarrow$ n bulat

Analisi kompleksitas :

1. Jumlah proses di exec . program
2. Time process

```
int   x = y+1 ;  → (1 operasi)
      y = x+2 ;  → (1 operasi)        } O(1)
      for (int i = 1 ; i <= N ; i++) {
            cout << i << endl;
      }
```

akan berulang sebanyak N x

N = 5 ⟶

```
cout << 1 << endl
cout << 2 << endl
cout << 3 << endl
cout << 4 << endl
cout << 5 << endl
```

} 5 volte

N ⟶

```
cout << 1 << endl
cout << 2 << endl
 -  -  -  -  -
cout << N << endl
```

} N x

Complexity: O(N)

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

$O(NM)$

Setiap i dia akan cetak j sebanyak M

i sebanyak N → operasi

sebanyak $N \times M$

$i = 1$
$j = 1$
$j = 2$
$j = 3$
...
$j = M$
$i = 2$
$j = 1$
$j = 2$
...
$j = M$

$f(n) = n \times f(n-1)$

$f(5) \rightarrow 5 \times f(4) \rightarrow 4 \times f(3) \rightarrow 3 \times f(2) \rightarrow 2 \times f(1) \rightarrow 1$

$f(n) = n \times (n-1) \times (n-2) \times \ldots \times 1$

$= 1 \times 2 \times 3 \times \ldots \times n$

Banyak operasinya : N

$O(N)$

$O(N)$ $\rightarrow 1 \leq N \leq 10^{10}$

1 detik $\rightarrow$ $N = 10^8$

$O(10^8 \cdot 10^2) \simeq$ 1 detik, ...

```
for (int i = 0; i < x; i++){
    cin >> y;
    cout << faktor(y) << endl;
}
```

x operasi

y operasi

untuk setiap i operasi ada O(xy)

karena i sebanyak x

$x = 7$, $y = 7$

$x = 10^7$, $y = 10^3$

$O(x \cdot y) = O(10^{10})$

10^8 = 1 detik

10^{10} = 1 detik lebih

TLE : Time limit Exceed

T = 5

4 → 4! = 1x3x2x1 = 24

4 → 4! = 4x3x2x1 = 24 } sama aja kurang efisien

4 → 4! = 1x3x2x1 = 24

4 → 4! = 4x3 = 24

4 dst.

T = 5

3 → 3! = 3 × 2 × 1 = 6

4 → 4! = 4 · 3! = 4 × 6 = 24

5 → 5 × 4! = 5 × 24 = 120

6 → 6 × 5! = 6 × 120 = 720

7 → _ _ _ .

3! = 6

Memoisation → Dynamic Programming (DP)

Memoization DP → O(N), O(log N)

O(log N) → $2^{\log N}$

O(log 10^{10}) → $2^{\log(10^{10})} \leq 10^8$

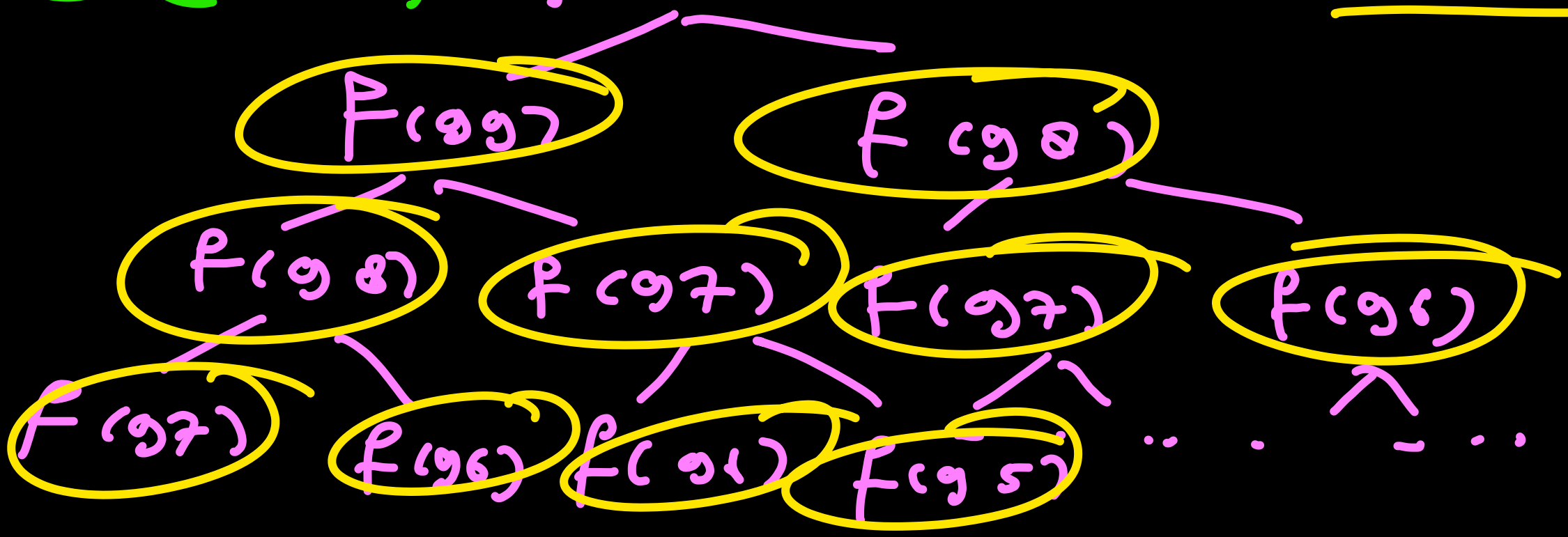
Problem solving

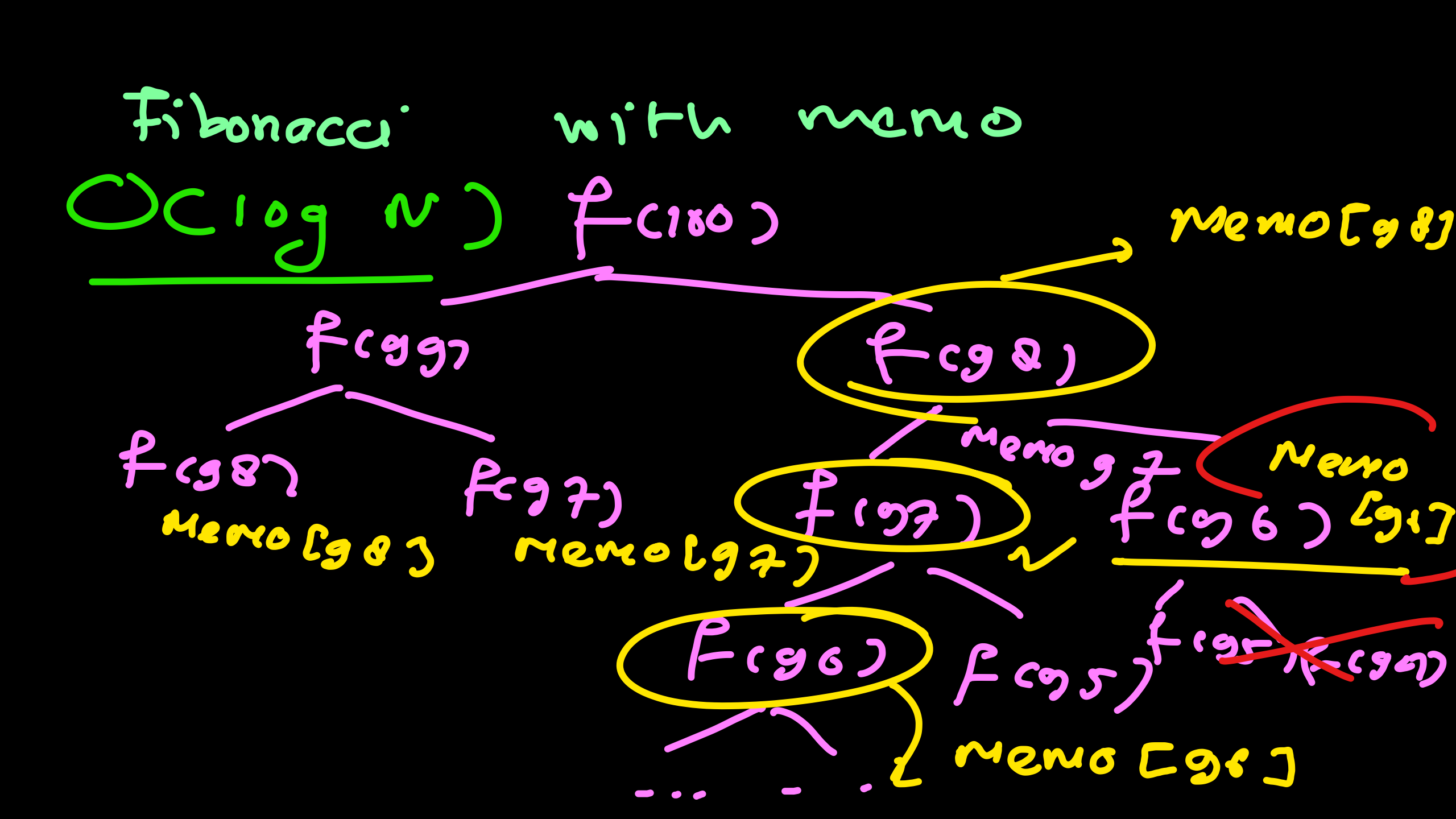
- Greedy
- Dnc
- Dp

Fibonacci without Memo

$O(2^N)$

f(100) → 2^{100} children




Fibonacci with memo
O(log N)
f(100)
f(99)
f(98)
memo[98]
f(97)
memo[97]
f(98)
memo[98]
f(97)
memo
f(96)
f(96)
f(95)
memo[96]
memo[95]

Without memo: $O(2^N)$

$N = 100 \longrightarrow 2^{100} \rightarrow 10^{30} \ldots 1e30$

Without memo: $O(\log N)$

$N = 100 \rightarrow (2 \log(100)) = [6, \ldots 7]$

$= 7$

OS mp $\leq$ 2s $\rightarrow 10^{16}$
256 mb

N x M $\rightarrow$ $1 \leq N, M \leq 10^5$

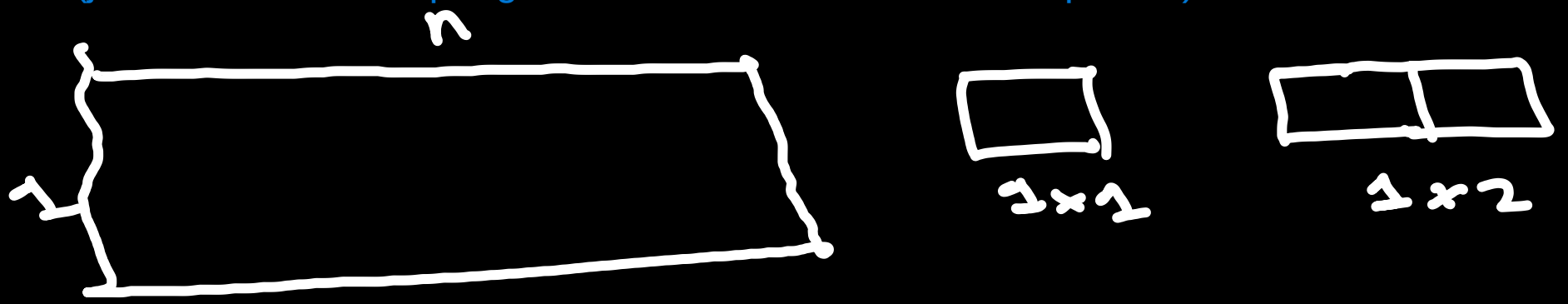
$\rightarrow O(NM)$

$\rightarrow O(10^5 \cdot 10^5) = 10^{10}$ (x)
$= 1, \ldots \leq 21$

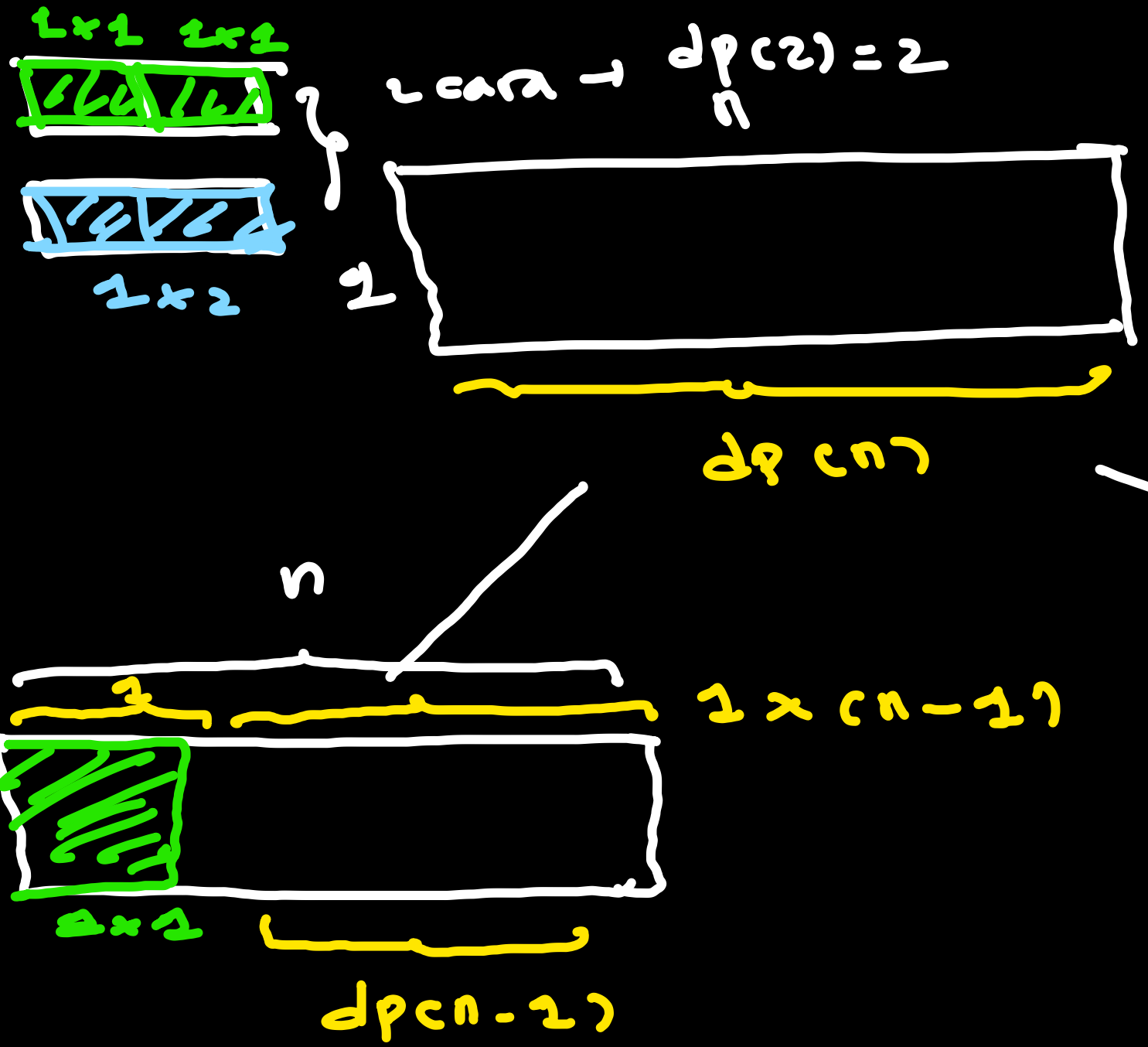
int jawaban = N % M → O(1)

Dynamic Programming

Misalkan kamu punya lantai berukuran 1 x N. Kamu
punya ubin berukuran 1 x 1 dan 1 x 2. Berapa banyak cara memasang ubin
(jika tidak boleh tumpang tindih dan semua lantai tertutupi ubin)



misalkan fungsi dp(n) = menyatakan banyak cara mengisi lantai berukuran 1 x n dengan ubin 1 x 1 dan 1 x 2

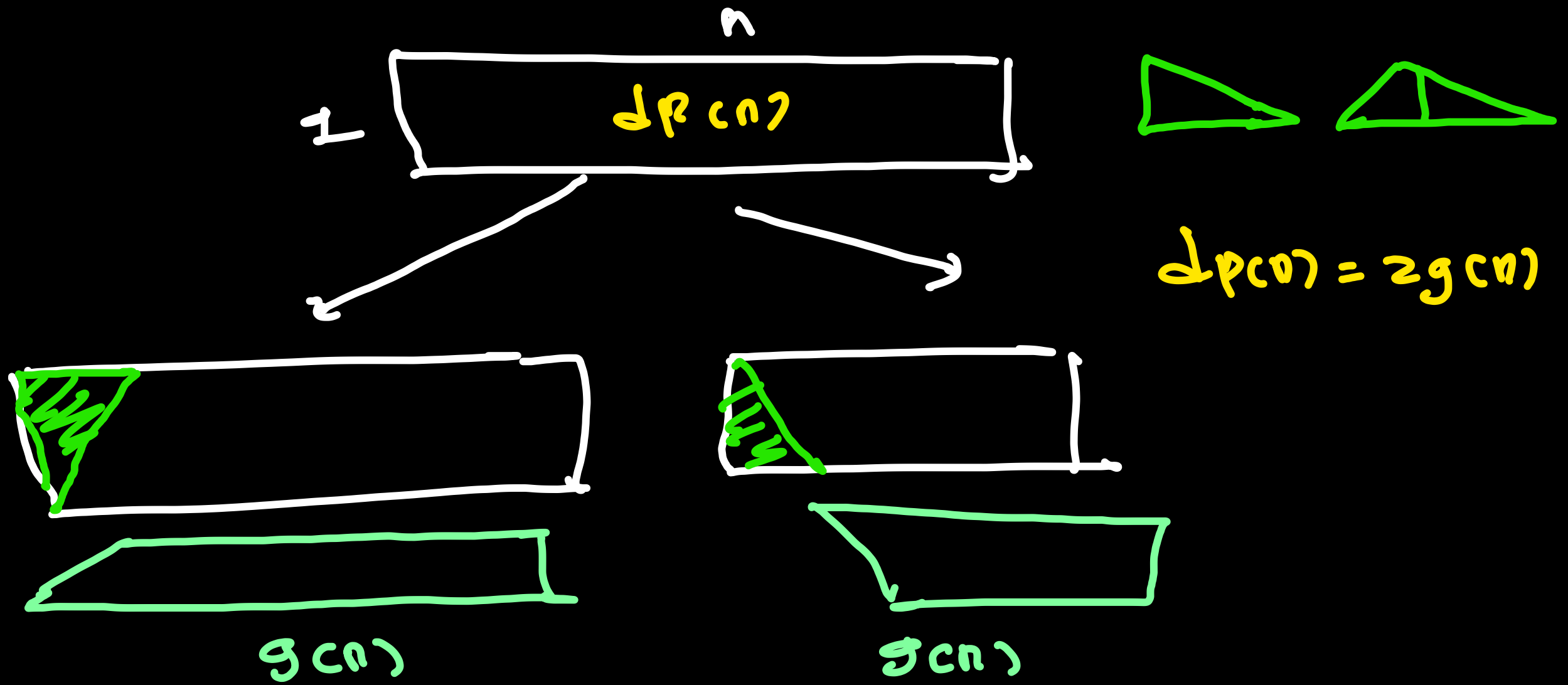


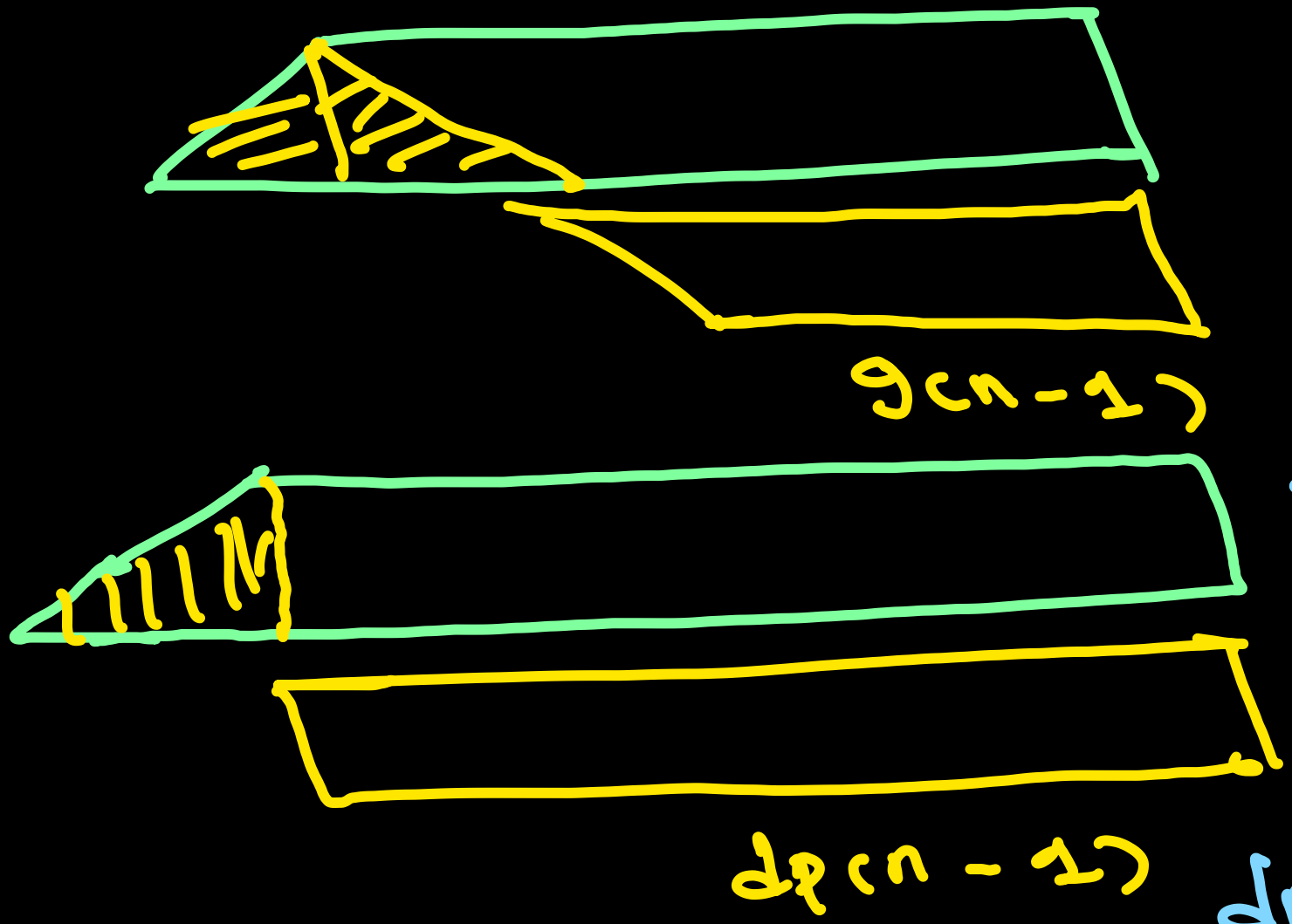
$dp(n) = dp(n-1) + dp(n-2)$

$dp(0) = 1$

$dp(1) = 1$

$dp(2) = 2$


n
1
dp(n)
dp(n) = 2g(n)
g(n)
g(n)



$$dp(n) = 2g(n)$$

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

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

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

$$g(n) = 3g(n-1)$$

$$dp(n) = 2 \cdot 3g(n-1)$$

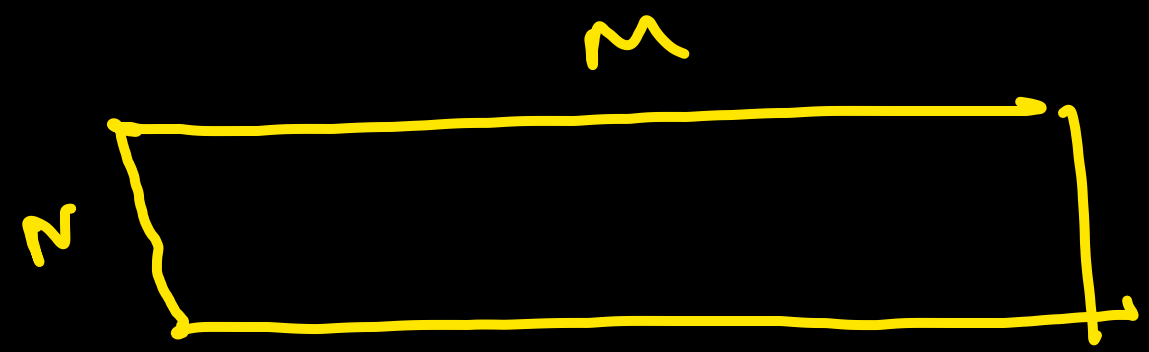
$$= 6g(n-1)$$

$$dp(n) = 2g(n)$$

$$dp(n-1) = 2g(n-1) \rightarrow 6g(n-1) = 3dp(n-1)$$

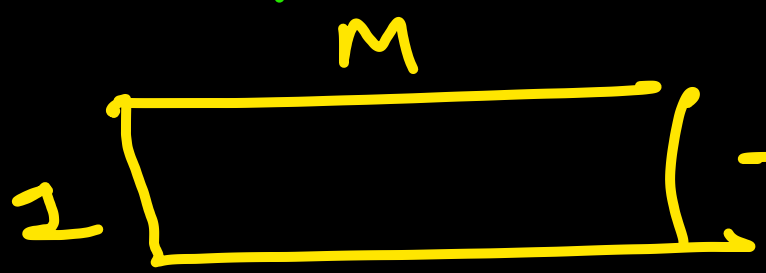
$$dp(n) = 3dp(n-1)$$

lantai dengan ukuran N x M



dp(n) = 3dp(n-1)

dp(M) = 3dp(M-1) → karan N = 1



```
cin>>N>>M;
if(N == 1){


cout<<dp(M)<<endl;

}
```

$dp(1) = 2$

$dp(M) = 3\,dp(M-1)$

$dp(2) = 3\,dp(2-1) = 3\,dp(1) \rightarrow 3.2$

$= 6$

$dp(3) = 3 \cdot 6 \rightarrow 3.(3.2)$

$= 18$

$dp(4) = 3 \cdot 18 \rightarrow 3(3.(3.2))$

$= 54$

$dp(M) = \underbrace{3 \times 3 \times 3 \times 3 \times \ldots \times 3}_{M-1} \times 2$

$$d(PCM) = \frac{3^{M-a}}{} \times 2$$

A = [5 6 3 9 2 7 9 5]

B = [7 5 5 7 3 9 3 6 5 3]

$R_A = 36/8$

$= 9,5$

$R_B = 48 / 10$

$= 9,8$

$R_A \not> R_B$

$R_A < R_B$

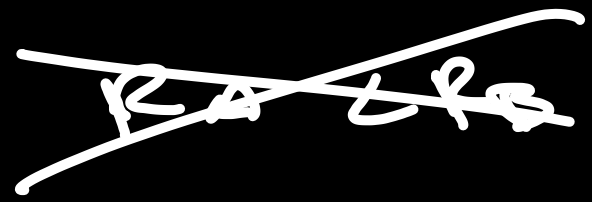
$R_A > R_B$

sesudah pemindahan

$R_A' = 36+3/9 = 39/9$

$= 9,3$

$$R_{B'} = 48 - 3/9$$
$$= 45/9$$
$$= 5$$



Kalau kita mindahin kentang yang bobot kecil di truk B ke truk A maka rata rata A akan berkurang dan rata B bertambah

Bagaimana caranya kita bikin rata rata A bertambah dan rata - rata B berkurang? sehingga tercapai rata rata A > rata rata B

-> Pindahkan kentang dengan bobot bobot yang besar ke truk A

A = [N kurang]

B = [M kurang]

A = [N+1]

B = [M-1]

B [1 kurang] → udah pindahin

B [M kurang] → B [1 kurang]

M+1 Pindahin tuk A

M - (M+1) = 1

A [N+M+1]

B.sort( )

B = [7 7 6 5 5 5 4 3 3 3]

for (i=0 ; i < B.length ; i++) {

B[i]

}