DP DP DP DP DP AAAAA

By Abdan Hafidz

Efisiensi : Analisis Kompleksitas

Time → TLE

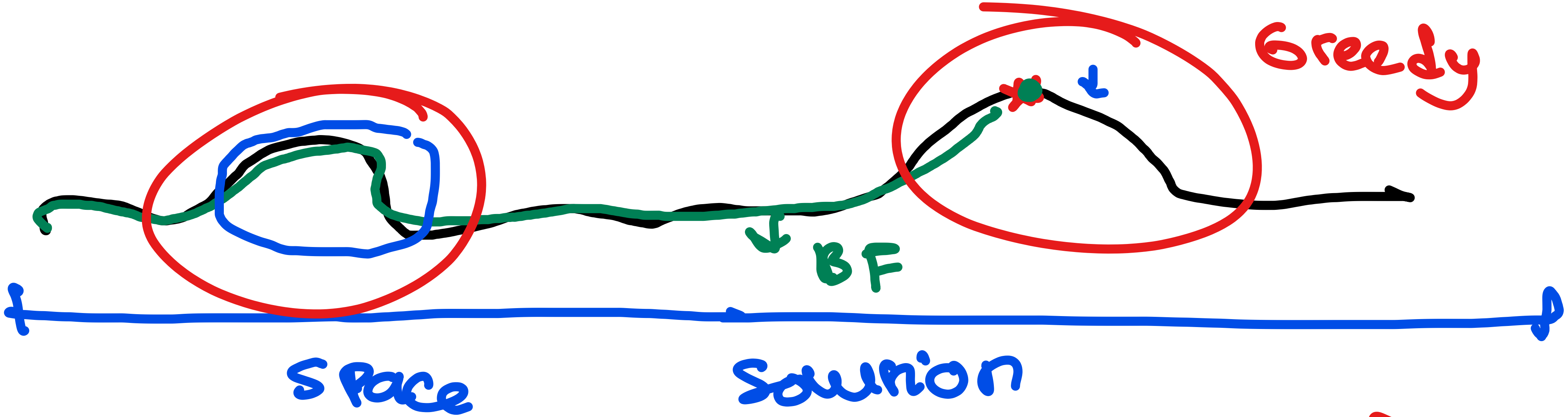
Space → MLE

Problem Solving Paradigm

- Brute - Force
- Greedy
- D n C – BSTA
- DP

D n C → Rekursif

* Kesesuaian solusi dgn konteks soal
* Kompleksitas ⇒ Cepat & Tepat



$$n! = n * f(n-1)$$

Soal DP / Rekursif :

1. Optimasi = Min - Max
2. Kombinatorika

* Recursive Intuition

Solusi untuk state sekarang = solusi state sebelumnya + B

$$f(a,b) = a + b$$

$\underbrace{a,b}_{\text{ekspresi}}$

① Base case

② Recurrence

↓ Transisi

* P. Recognition

$$f(a,b) = a + b$$

$$f(5,3) = \underbrace{5 + 3}_{} = (5 + 2) + 1$$

$$8 \qquad 7 + 1 \qquad f(5,2) \qquad 8$$

$$f(\overline{5,3}) = f(\overline{5,2}) + 1$$

7 Transition 6 + 1

$$f(\overline{5,2}) = f(\overline{5,1}) + 1$$

Transition

$$f(\overset{5}{a},\overset{3}{b}) = f(\overset{5}{a},\overset{2}{b-1}) + 1$$

$$f(a,b) = a + b \rightarrow f(a,\underbrace{b-1}_{\text{transisi}}) + 1$$

depth Recursion limit

base case

$$f(a,0) = a \quad \text{(base case)}$$

Bentuk rekursif dari

1) f(a,b) = a - b, f(a,b) = f(a,b - 1) - 1
2) f(a,b) = a * b, f(a,b) = f(a, b - 1) + a
3) f(a,b) = a / b, f(a,b) = f(a, b - 1) - a
4) f(a,b) = pow(a,b), f(a,b) = f(a, b - 1) * a

1) f(a,b) = a - b, f(a,b) = f(a,b - 1) - 1

2) f(a,b) = a * b, f(a,b) = f(a, b - 1) + a

3) f(a,b) = (int) a / b, f(a,b) = f(a, b - 1) - a

4) f(a,b) = pow(a,b), f(a,b) = f(a, b - 1) * a

Base case

f(0) = ...

f(1) = ...

$$f(a,b) = a/b$$

B. case

$$f(a,1) = a$$

$$f(a,b) = f(a, \underset{8}{b-1}) - a$$

$$f(8,2) = \overbrace{f(8,1)} - 8 = 0$$

$$f(8,1) = 8$$

$$15 : 3 = 15 \underbrace{\overbrace{-3-3-3-3-3}^{=0}}_{5 \text{ kali}}$$

$$a : b = a - b - b - b - \dots - = 0$$

$$f(a, b) = f(a-b, b) + 1$$

$$f(0, b) = \boxed{0}$$

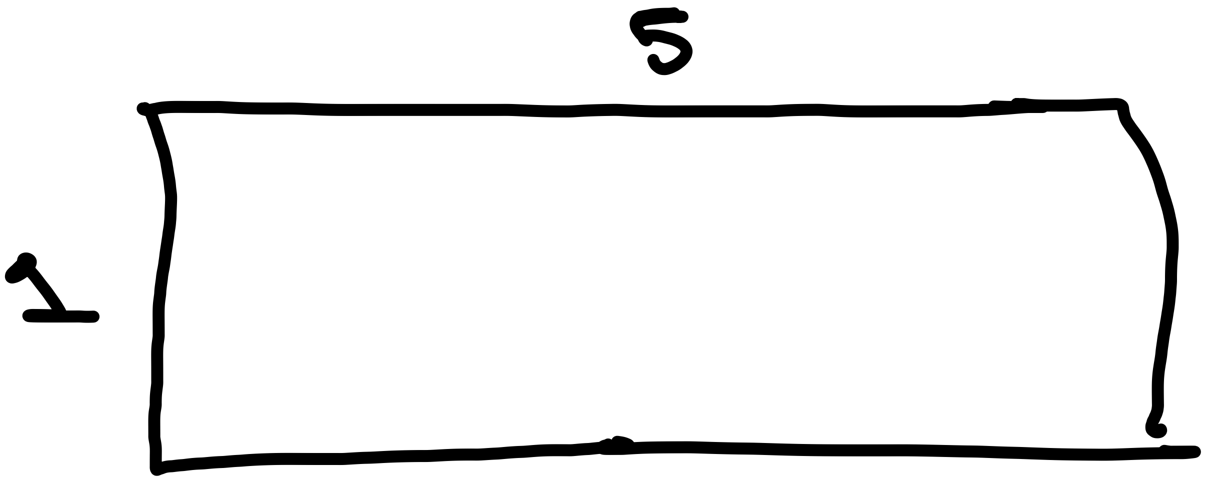
$15 : 3 = 5$

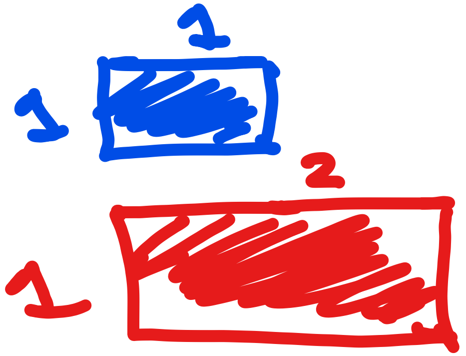
$$f(15,3) \to f(12,3) \to f(9,3) \to f(6,3) \to f(3,3) \to f(0,3)$$

Diberikan lantai berukuran 1 x 5, Pak Justin ingin memasang ubin berukuran 1 x 1 dan 1 x 2 (Pemasangan tidak boleh tumpang tindih dan ubin dapat dirotasi ataupun refleksi)

P. Panjang p l

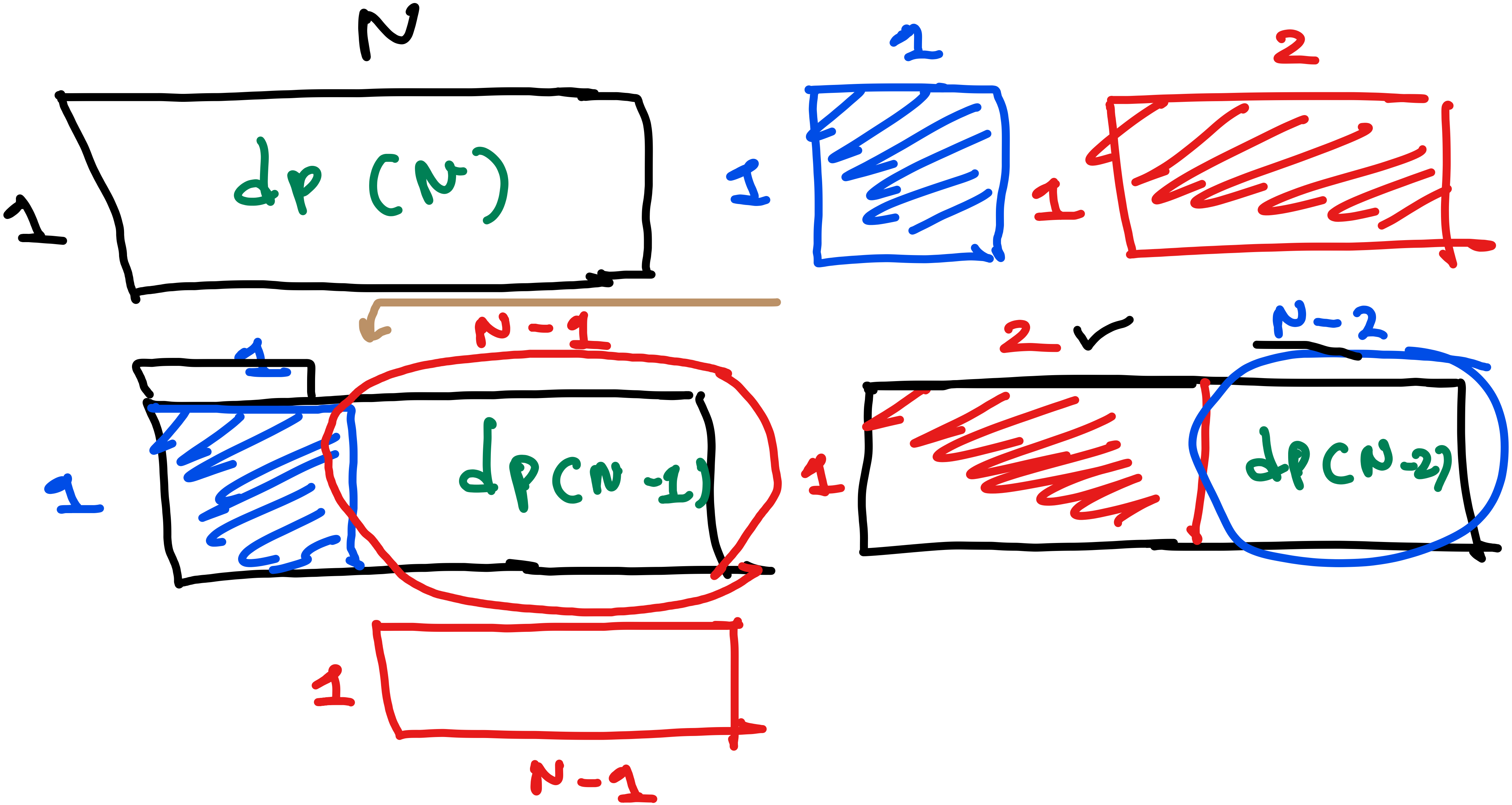
Ada berapa banyak cara yang bisa dilakukan?



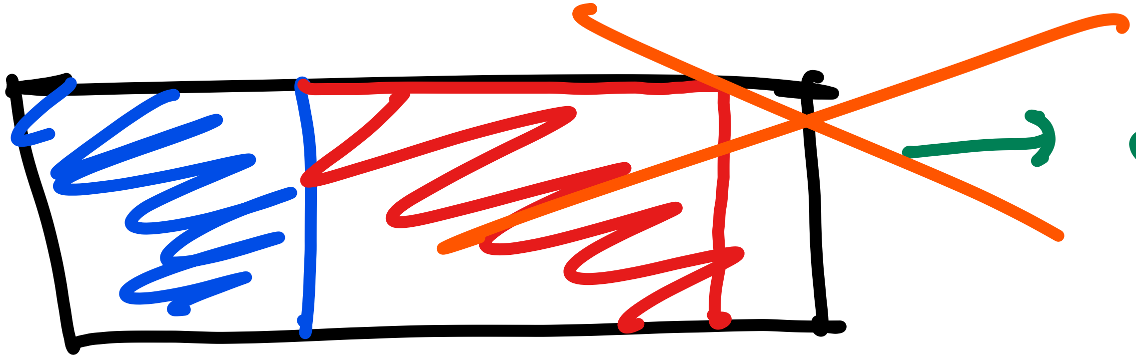


1) State function rekursif (DP State)

$dp(n) =$ banyak cara mengisi lantai berukuran 1 x N

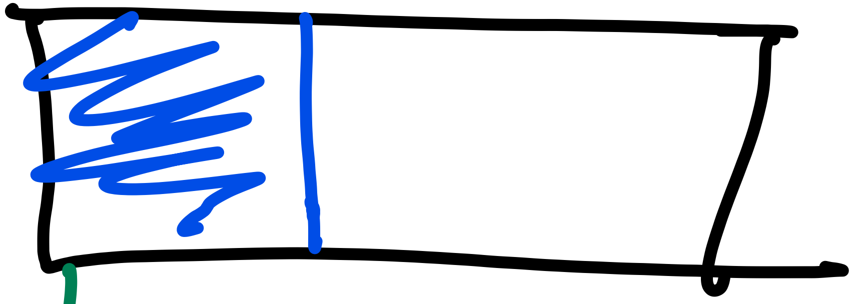


N
1
dp (N)
1
1
2
1
N-1
1
1
dp(N-1)
1
2 ✓
N-2
1
dp(N-2)
1
N-1



$dp(N-3)$

bagian dari



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

$$dp(N) = dp(N-\boxed{1}) + dp(N-\boxed{2})$$

orde orde

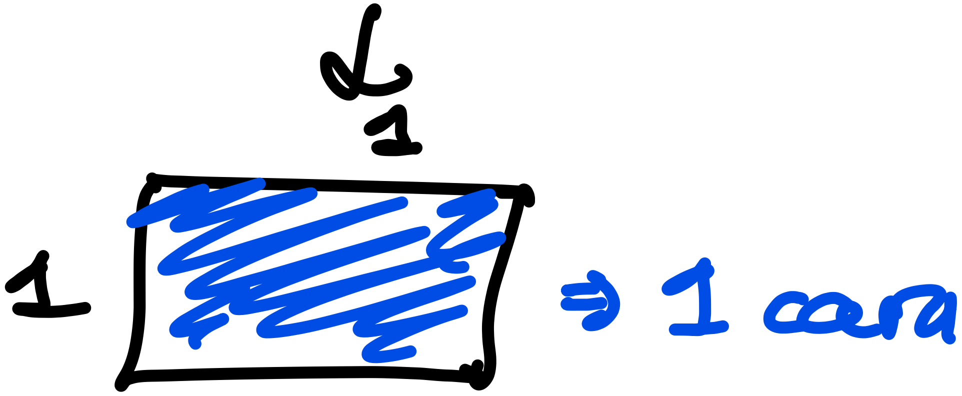
$$f(N) = f(N-1) + f(N-2) + \dots + f(N-k)$$

orde Max = k

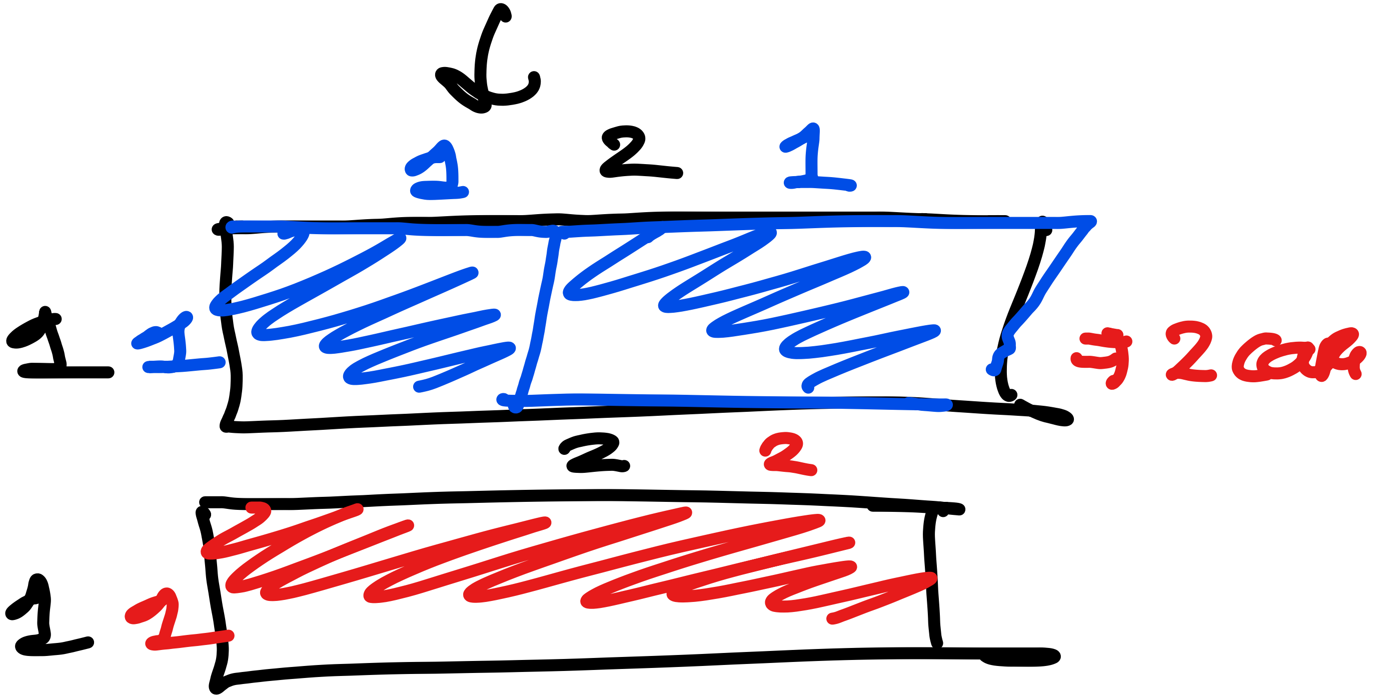
Base case $\rightarrow$ $f(0)$ s.d $f(k-1)$

$f(1)$ s.d $f(k)$

$dp(1) = 1 \ldots$



$dp(2) = 2 \ldots$



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

$dp(1) = 1$ $\qquad$ $dp(2) = 2$

$dp(0) = 1$ — $0! = 1$, $a^0 = 1$

1×5 $\qquad$ ans $= \underline{\underline{dp(5)}}$

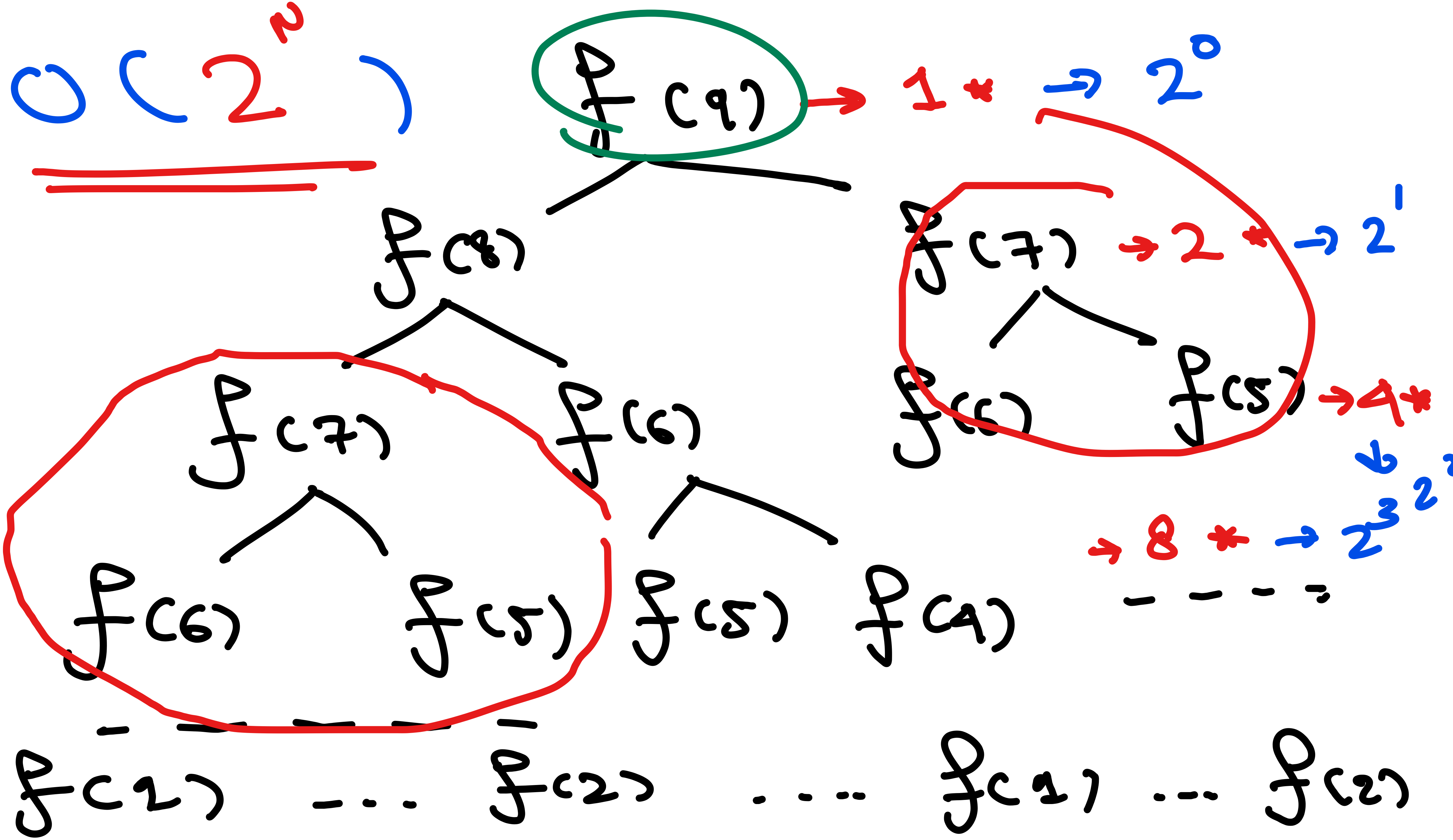
$$f(N) = f(N-1) + f(N-2)$$

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

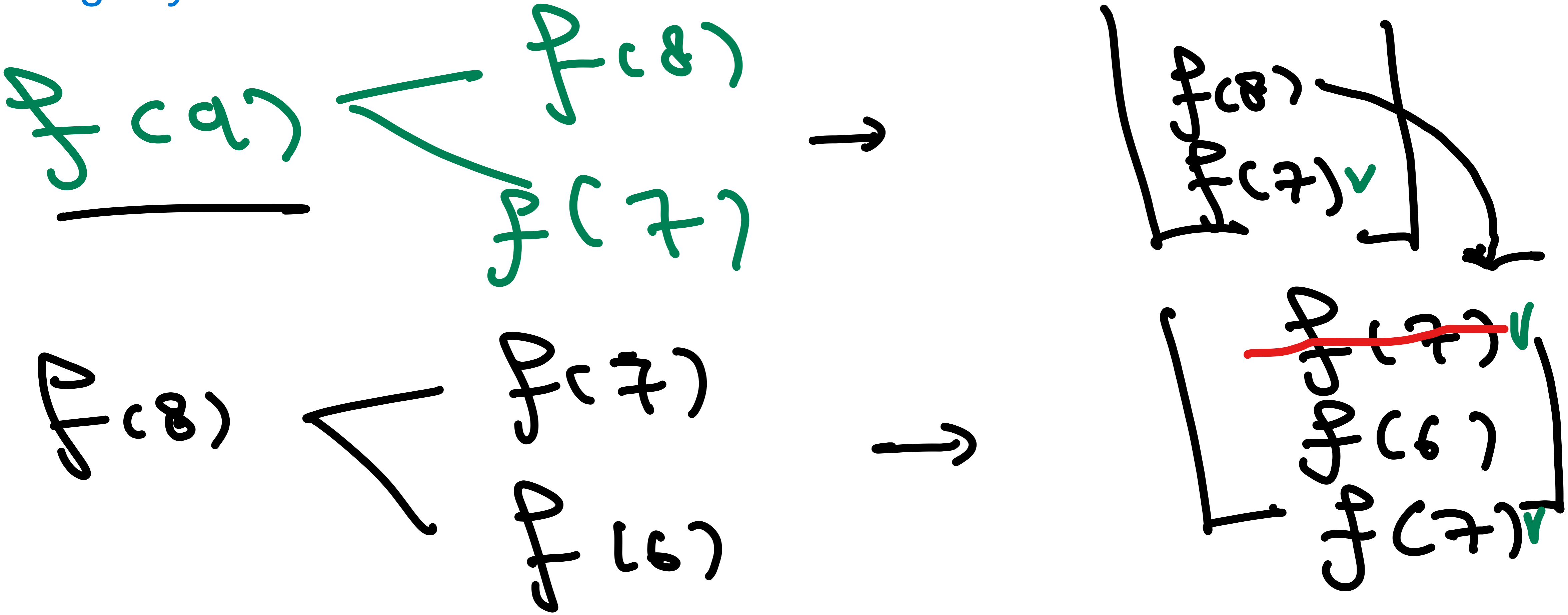
$10^8 \to 1$ denik

$2^{100} \to > 1$ deni

$O(2^N)$



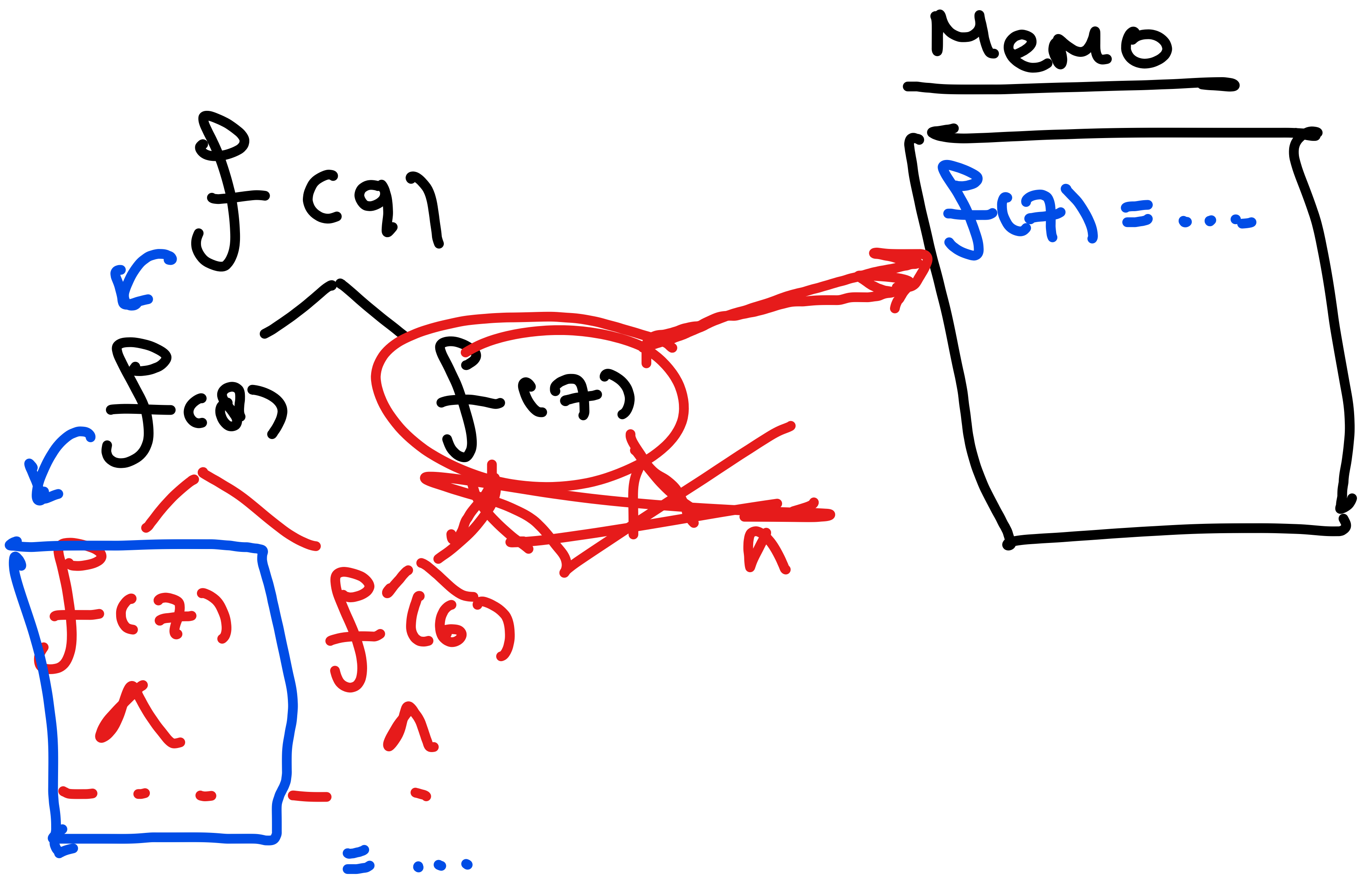
Kompleksitas sebuah fungsi rekursif => Berdasarkan pemanggilan fungsinya

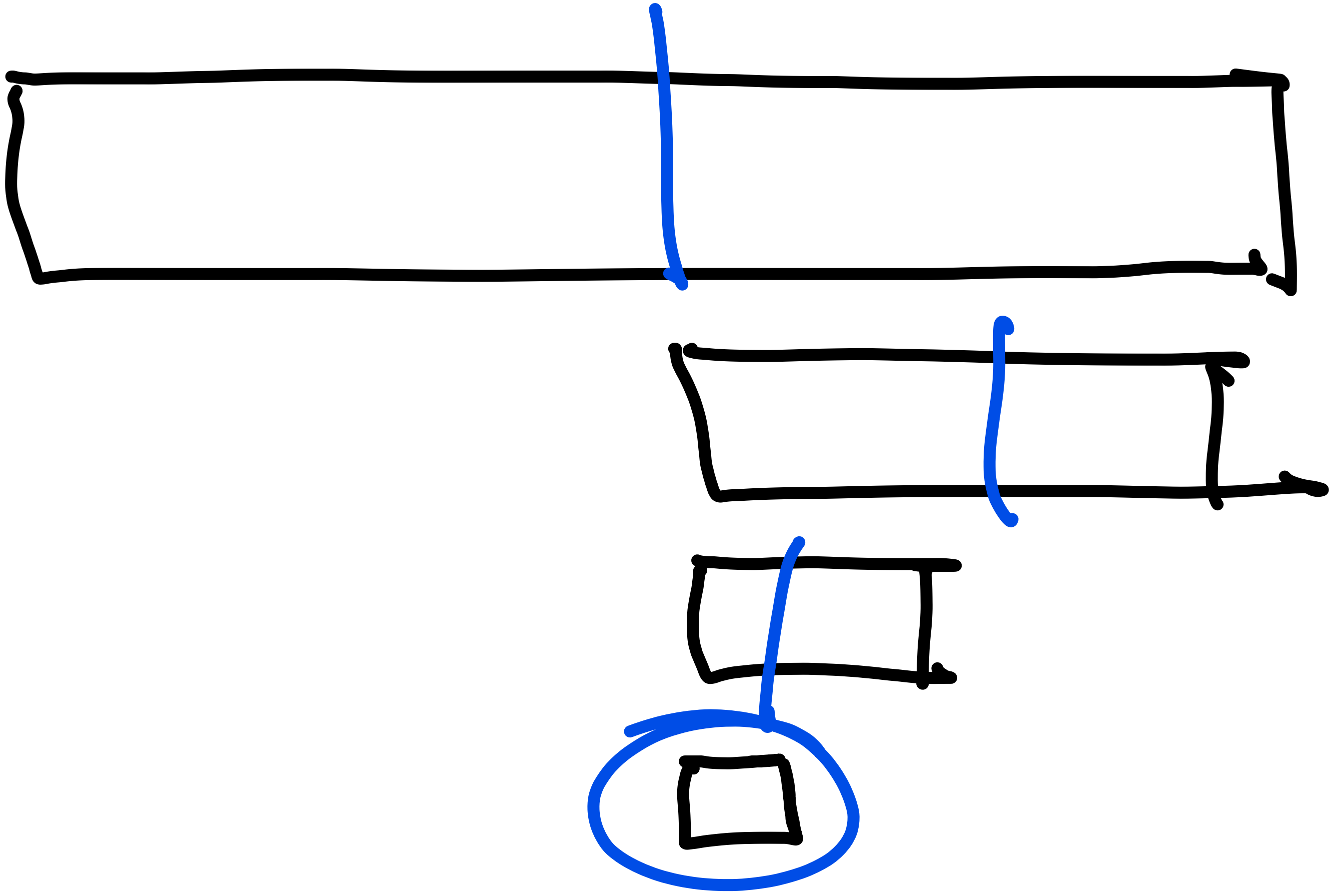


DP ⇒ Good Time complexity

Trade of Memory comp..

memo[n] = dp(n)


Memo
f(7) = ...
f(9)
f(8)
f(7)
f(7)
f(6)
= ...



Coin Changes

Max val ⇒ O(log N)

Kita mau membayar untuk pembelian suatu barang dengan harga N, jika kita punya M pecahan koin A1, A2, A3, ... , AM (max(AM) == N).

↳ O(N)

Berapa jumlah koin minimal yang diperlukan untuk membayar belanjaan tsb

N = 7000

A_3

koin = A = { 100 , 200 , 500 } ✓

7000 = 100 100 100 --- 100

70 koin

$7000 = \underbrace{\underbrace{100 \quad 100 \quad 100 \quad \ldots \quad 100}_{68 \text{ koin}} \quad \underbrace{200}_{1 \text{ koin}}}_{79 \text{ koin}}$

Min = 14 → 7000/500 → greedy

* greedy

Jml Pecahan koin = Σ harga / A_i

x / y = min

y → max ⇒ greedy

Belanjaan seharga N

Dan punya koin pecahan A_i $(1 \le i \le m)$

Jml pecahan = harga/pecahan

$$dp(N) = \min(dp(N-A_1), dp(N-A_2), dp(N-A_3), \dots, dp(N-A_i)) + 1$$

$$dp(N) = \left(\min_{(1 \le i \le M)} \underbrace{dp(N - A_i)}_{\text{koin mana yang bakalan menghasilkan jml pecahan min}} \right) + 1$$

Base case

Rec = 100, 200, 500

$$dp(N) = \underline{\text{Min}}\big(dp(N-\underline{100}),\ dp(N-\underline{200}),\ dp(N-\underline{500}) \big) + 1$$

$N - A_i \geq 0$

$N = 100$ $A_1 = 500$

Tdk mungkin $\underline{N - A_1} = 100 - 500 < 0$

$dp(N - A_1)$ (Gamau)

Base Case

$$dp(N) = \underline{\infty} \quad \text{if } N - A_i < 0$$

< 0

$dp(A_i) = 1$

2 koin 1 koin 60

$\underline{100}$, $\underline{200}$, $5\underline{00}$

A_1 A_2 A_3

$N ==$ salah satu A

$dp(N) = 1$

$dp(1\underline{00}) = 1$

$dp(200) = 1$

$dp(500) = 1$

$$dp(N) = \min(dp(N-100),\ dp(N-200),\ dp(N-500)) + 1$$

$$dp(A_i) = 1$$

$$\underline{dp(100)}_{N = A_i} = \min(dp(0),\ dp(-100),\ dp(-400)) + 1$$

$$dp(0) = 0$$

$$\infty \qquad \infty$$

$$dp(100-100)$$

$$dp(100) = 1$$

[Memasang Paku Bumi 24 – 25]

Tiga sekawan yaitu Pak Chanek, Pak Dengklek, dan Pak Ganesh akan memasang Paku Bumi yang sangat besar. Pak Dengklek dalam satu kali pukulan dapat membuat Paku tertancap sedalam 1 meter, Pak Chanek dalam satu kali pukulan dapat membuat Paku tertancap 2 meter, dan Pak Ganesh dalam satu kali pukulan dapat membuat Paku tertancap 5 meter.

Ketiganya dapat bekerja sama dengan sangat baik, mereka melakukan pemasangan Paku secara bergantian dan berirama, namun mungkin saja untuk memasang sebuah Paku tidak perlu semua orang bekerja.

Base case $= dp(0) = 1,\ dp(1) = 1$

24. Ketiganya memasang Paku berukuran 8 meter ada berapa banyak cara pemukulan Paku dengan urutan berbeda? {tuliskan jawaban dalam bentuk ANGKA saja}

D → 1M C → 2M G → 5M

25. Berapa ukuran Paku minimal sehingga ketiganya dapat bekerja memasang Paku secara bergantian? {tuliskan jawaban dalam bentuk ANGKA saja}

$dp(N) = dp(N-1) + dp(N-2) + dp(N-5)$

$dp(N)$ = Banyak cara pemasangan paku berukuran N meter

N meter

5M → D 1 C 2

1M 2M C 2M C

D CC

D 1M N-1 C 2M N-2 G 5M N-5

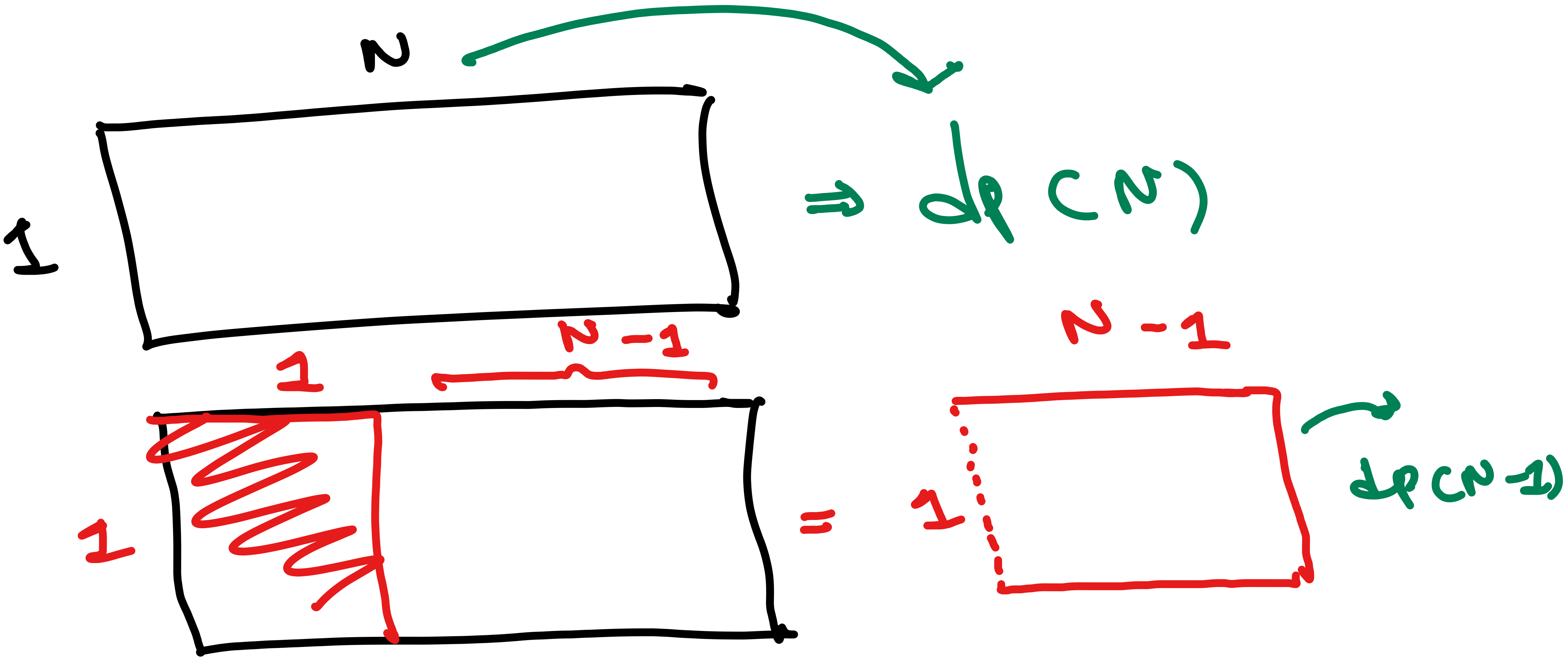
6 10 + 2C + 5G = P 8

15 1 1 1

Pak Dengklek ingin pindah rumah dan akan memindahkan N buah barang ke rumah barunya, Pak Dengklek mempunyai M buah cara mengangkut barang yaitu mengangkut A1 barang sekaligus, A2 barang sekaligus, ..., AM barang sekaligus dimana dijamin Ai >= 1,

Pak Dengklek orangnya pemalas, berapa jumlah langkah minimum yang akan dilakukan Pak Dengklek untuk memindahkan barang - barangnya tersebut ke rumah barunya?

dp(N) = ...

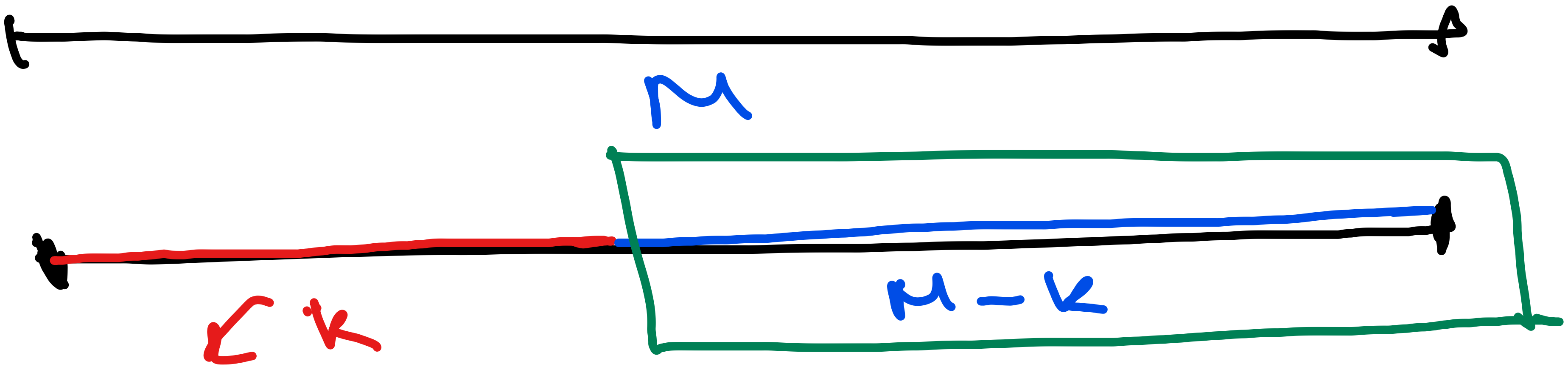

N
1
⇒ dp(N)
1
N - 1
1
=
N - 1
1
dp(N-1)

Masalah Besar (M)

Temukan sedikit bagian dari solusi M sebesar k.

Selesaikan sub problem yg tersisa

$M - k$



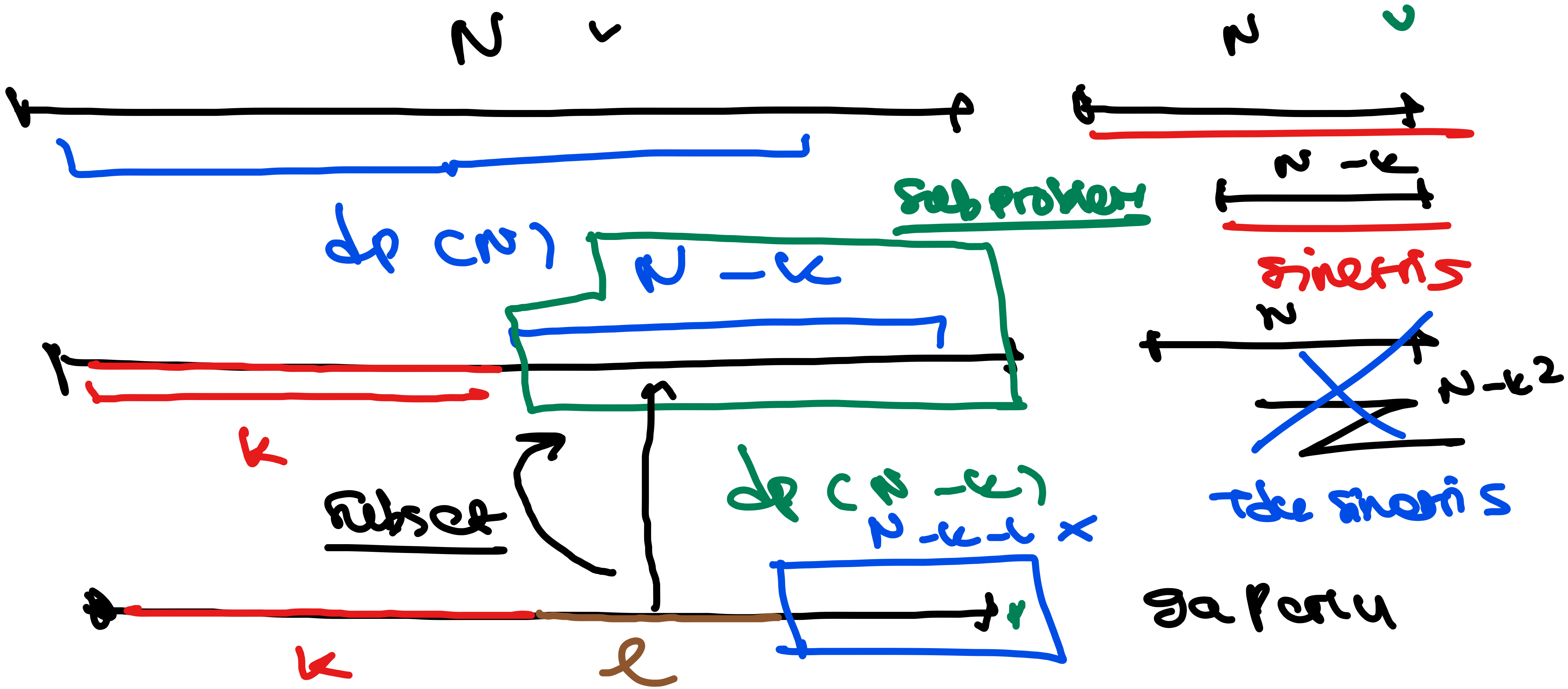
K is part of M
solved

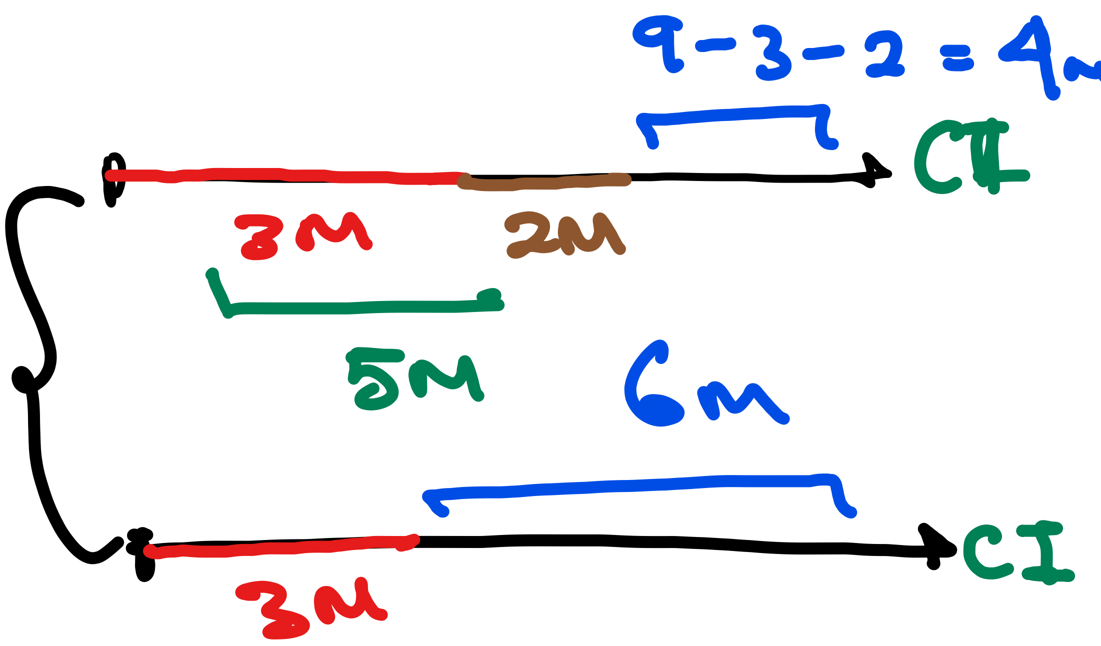
Transition state

Found Solution → l Ternyata

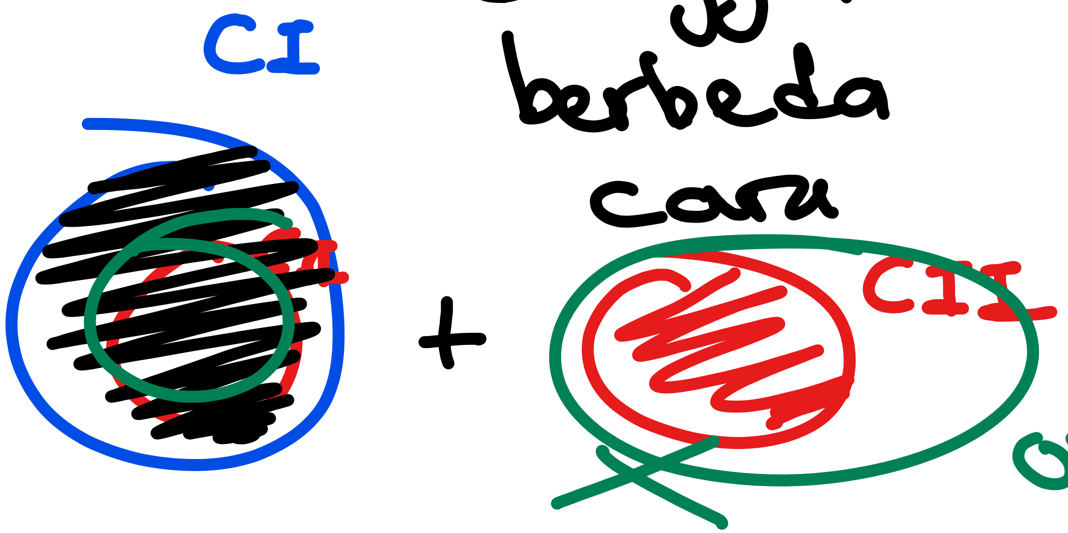
l part dari M − K

Tidak perlu dilanjutkan


N
dp (N)
N - k
subproblem
k
subset
dp (N - k)
N - k - 1
N - k



Jangan dianggap berbeda cara

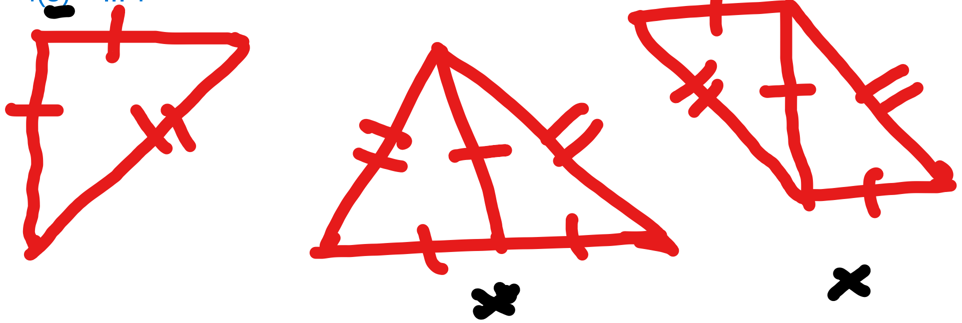


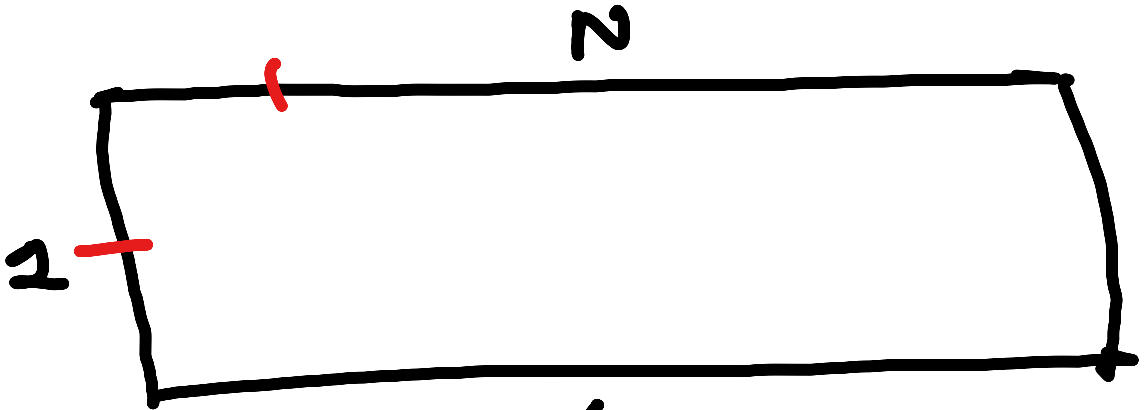
CII bagian dari CI

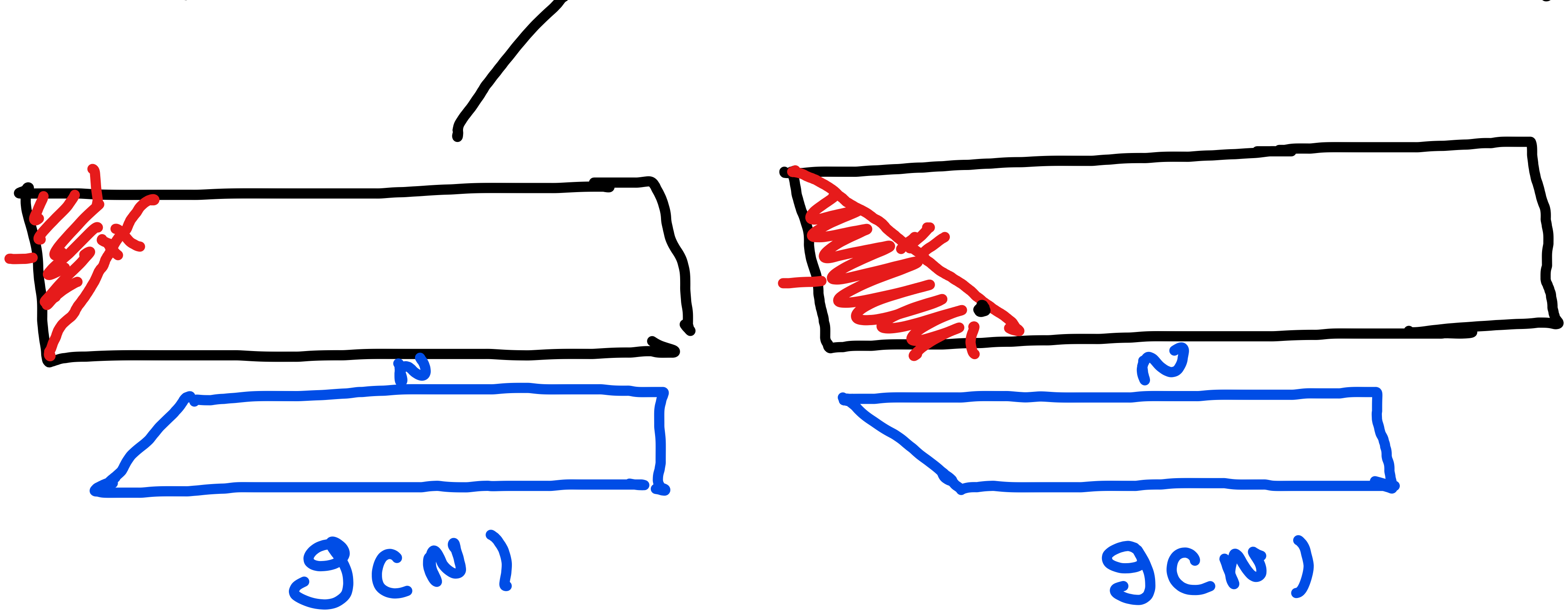
$f(\underline{N}) = \ldots$

f(N) = banyak cara menutup lantai berukuran 1 x N

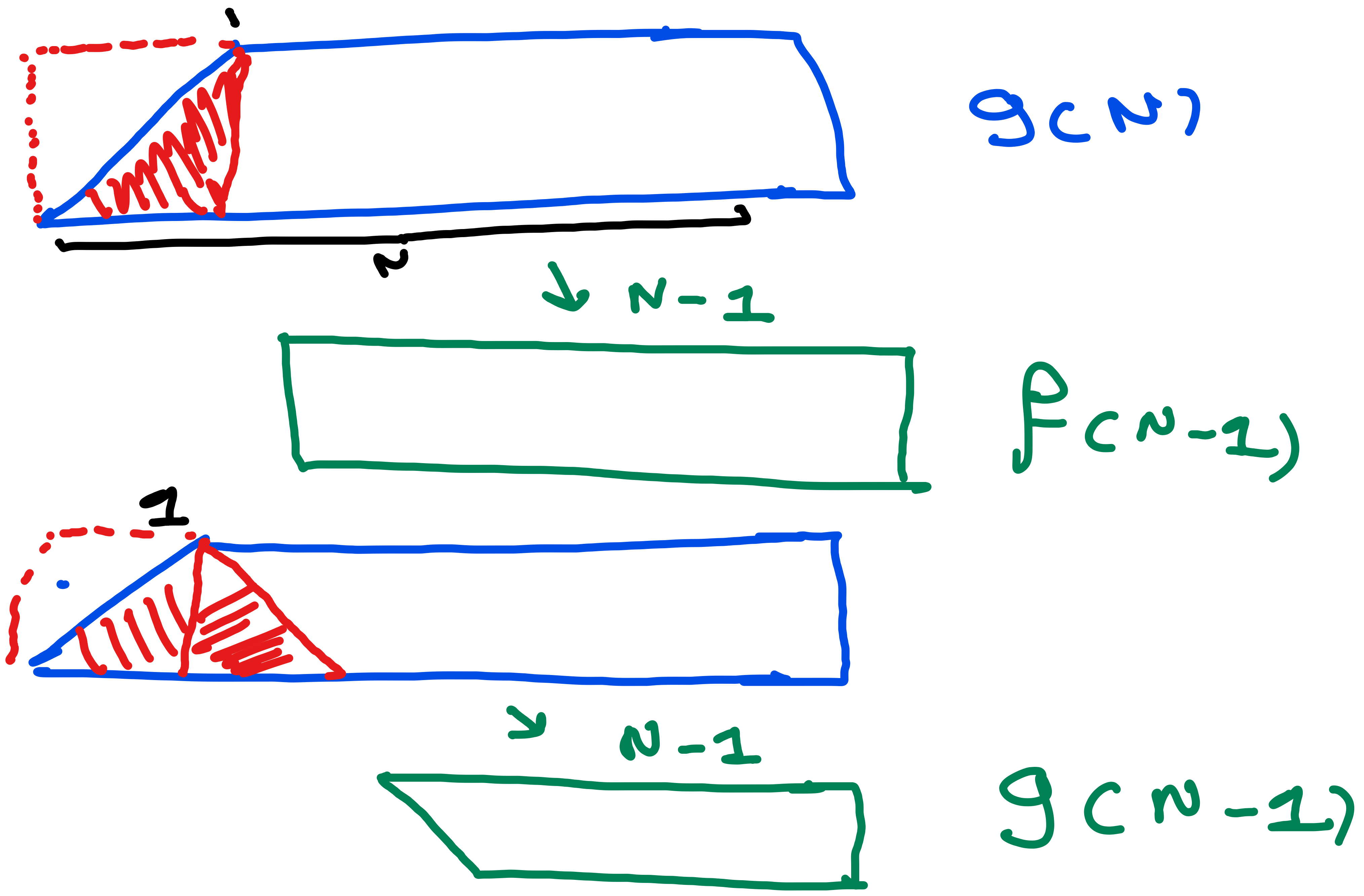
f(5) = ... ?



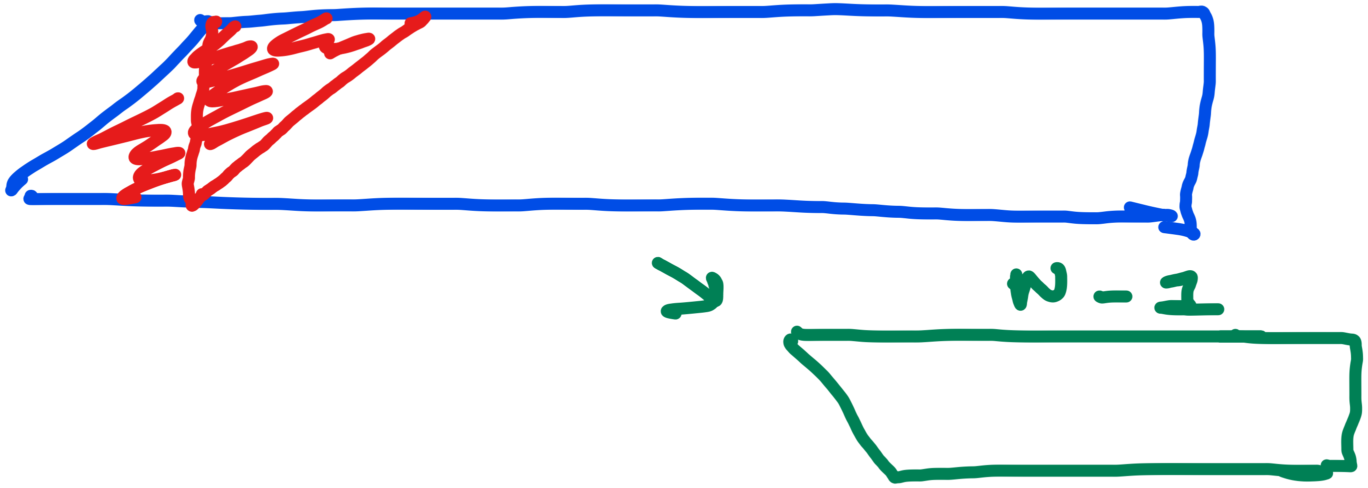




$$f(N) = 2g(N)$$



1
n
g(n)
n-1
f(n-1)
1
n-1
g(n-1)



$g(n-1)$

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

$$f(N) = \underline{2g(N)}$$

$$g(N) = f(N-1) + \underline{2g(N-1)}$$

$$2g(N) = f(N) \qquad f(N-1)$$

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

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

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

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

$$f(n) = 2 \cdot 2f(n-1)$$

$$\underline{f(N)} = \underline{4 f(N-1)}$$

$$f(0) = 1$$

$$f(1) = \underline{\underline{2}} = 2 \rightarrow 2^1$$

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

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

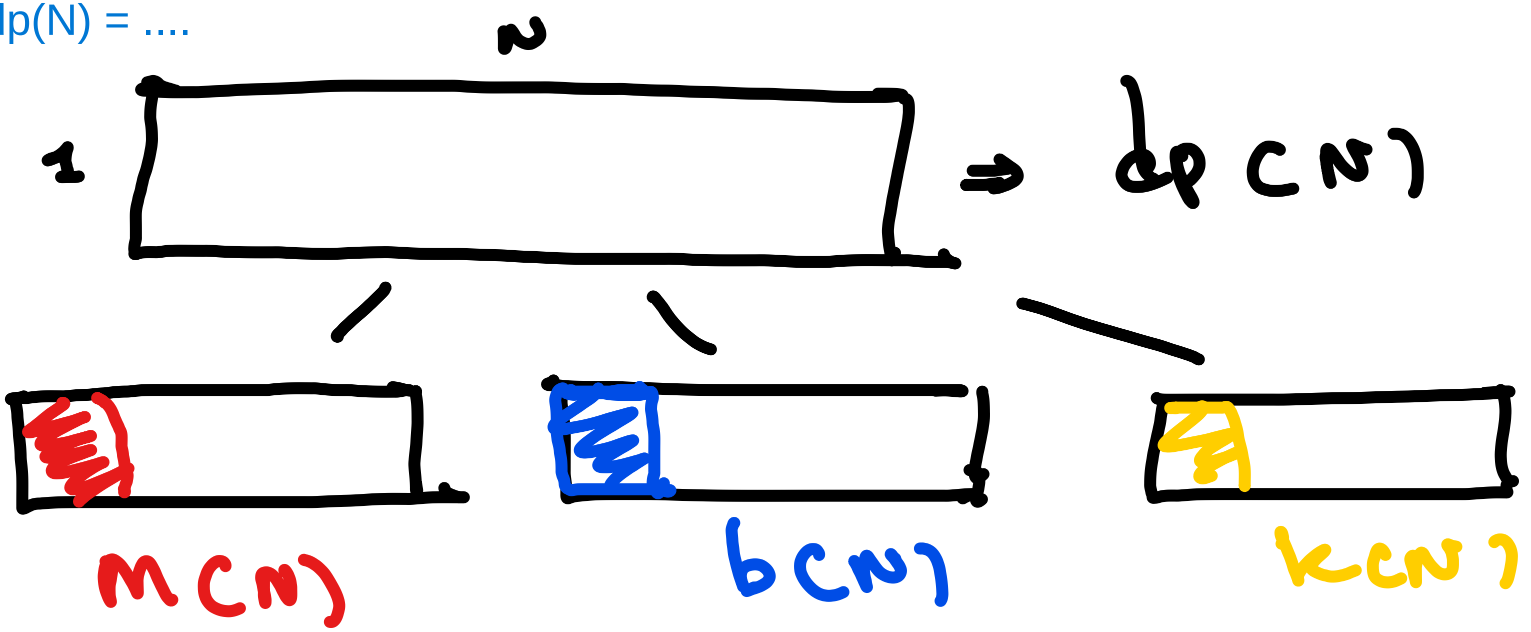
$$f(4) = 4 \cdot f(3) = \rightarrow 2^7$$

$$f(5) = 2^9 = \boxed{512}$$

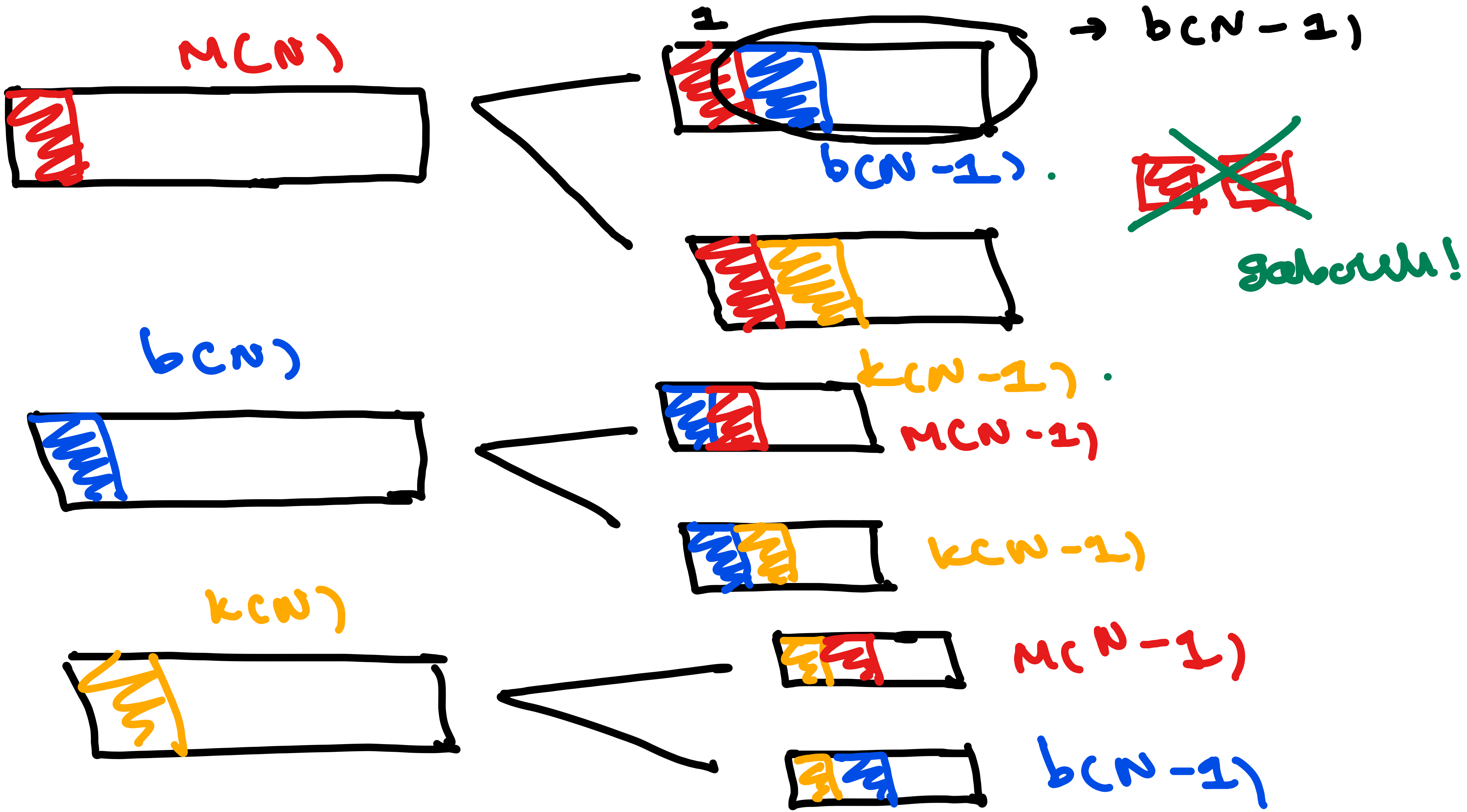
Kikik mempunyai lantai berukuran 1 x N, Kikik ingin mewarnai setiap ubin 1 x 1 dengan warna merah, biru, kuning.

Ada berapa banyak cara yang bisa dilakukan Kikik jika "Tidak boleh petak yang bersebelahan / bersisian berwarna sama"?

dp(N) = ...



$$dp(N) = m(N) + b(N) + k(N)$$



M(N)
1
→ b(N-1)
b(N-1).
k(N-1).
b(N)
M(N-1)
k(N-1)
k(N)
M(N-1)
b(N-1)

$$\underline{f(N)} = \underline{M(N)} + \underline{B(N)} + \underline{K(N)}$$

$$M(N) = B(N-1) + K(N-1)$$

$$B(N) = K(N-1) + M(N-1)$$

$$K(N) = M(N-1) + B(N-1)$$

$$+$$

$$f(N) = B(N-1) + K(N-1) + K(N-1) + \underline{M(N-1)} + \underline{M(N-1)} + B(N-1)$$

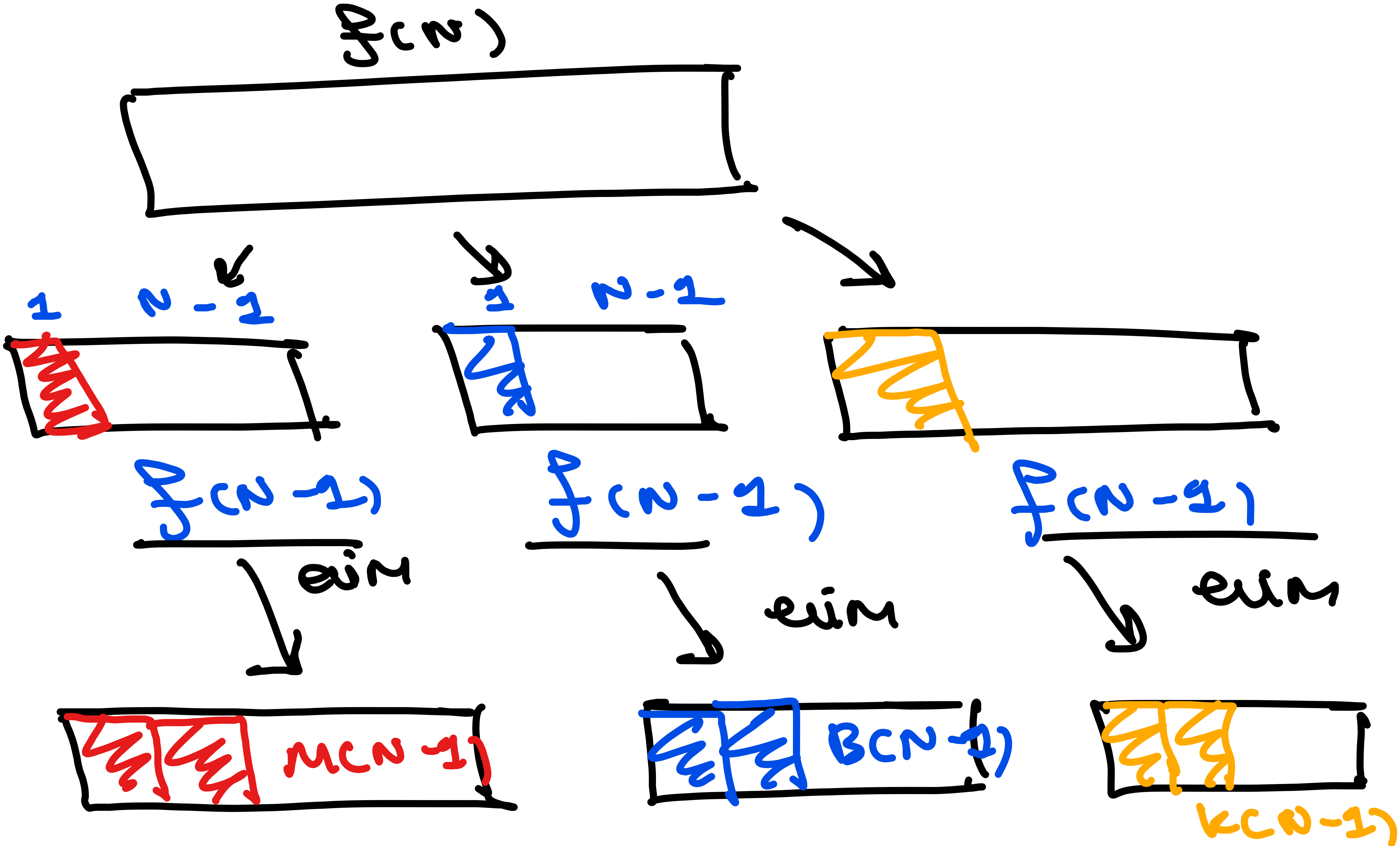
$$f(N) = 2M(N-1) + 2B(N-1) + 2K(N-1)$$

$$f(N) = 2\left(M(N-1) + B(N-1) + K(N-1)\right)$$

$$f(N) = M(N) + B(N) + K(N)$$

$$f(N-1) = \underline{M(N-1) + B(N-1) + K(N-1)}$$

$$f(N) = \underline{\underline{2f(N-1)}} \qquad f(1) = 3$$



f(N)
1
N-1
1
N-1
f(N-1)
f(N-1)
f(N-1)
sum
sum
sum
M(N-1)
B(N-1)
K(N-1)

$$f(N) = 3f(N-1) - \underbrace{(M(N-1) + B(N-1) + K(N-1))}_{f(N-1)}$$

$$f(N) = 3f(N-1) - f(N-1)$$

$$= 2f(N-1)$$

$$\text{DP} \begin{cases} \to \text{Bottom-up} \\ \to \text{Top-down} \end{cases}$$

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

Bottom-up

BC

$f(1) = 1$

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

$f(3) = 2 \cdot f(2) = 4$

$f(N) = \ldots$

$f(5)$

$\downarrow$

$f(4)$

$\downarrow$

$f(3)$

$\downarrow$

BC

Top down

$\downarrow$

Func

Recursion

Bottom-Up :

```
vector<int>dp;
dp[0] = 1
for(int i = 1; i<=N; i++){
   dp[i] = 2*dp[i - 1];
}
```

Top Down :

```
int dp(int n){
   return memo[n] = 2*dp(n - 1)
}
```

Dp on Grid

* Tabular

dp(N)

↳ 1 dimensi

$dp(x, y)$

$\longrightarrow$ 2 <u>dimensi</u>