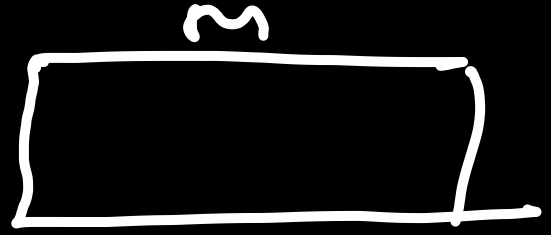
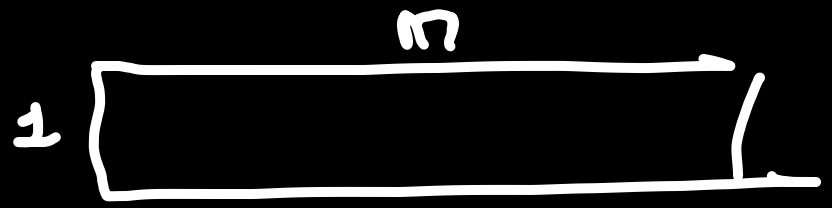
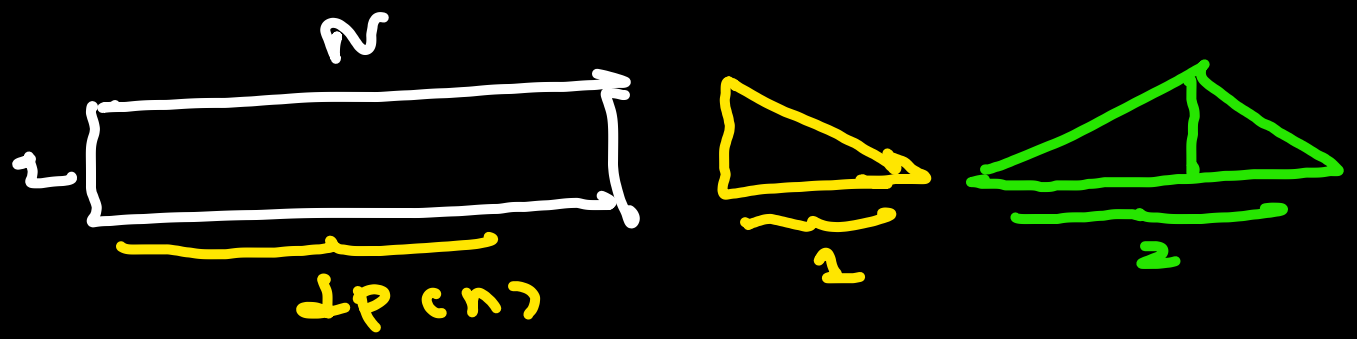
$N \times M$

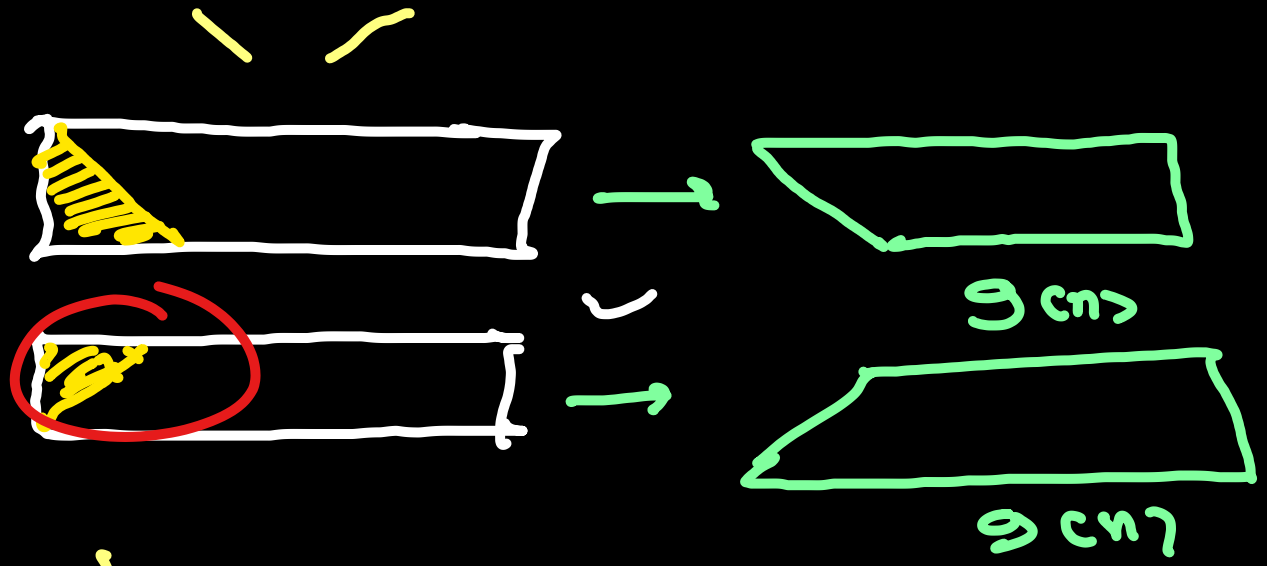


subsoal (N = 1) (80%)

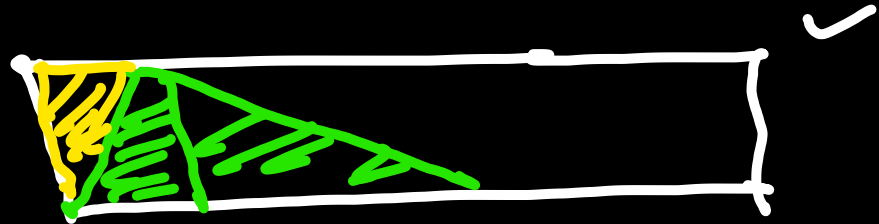




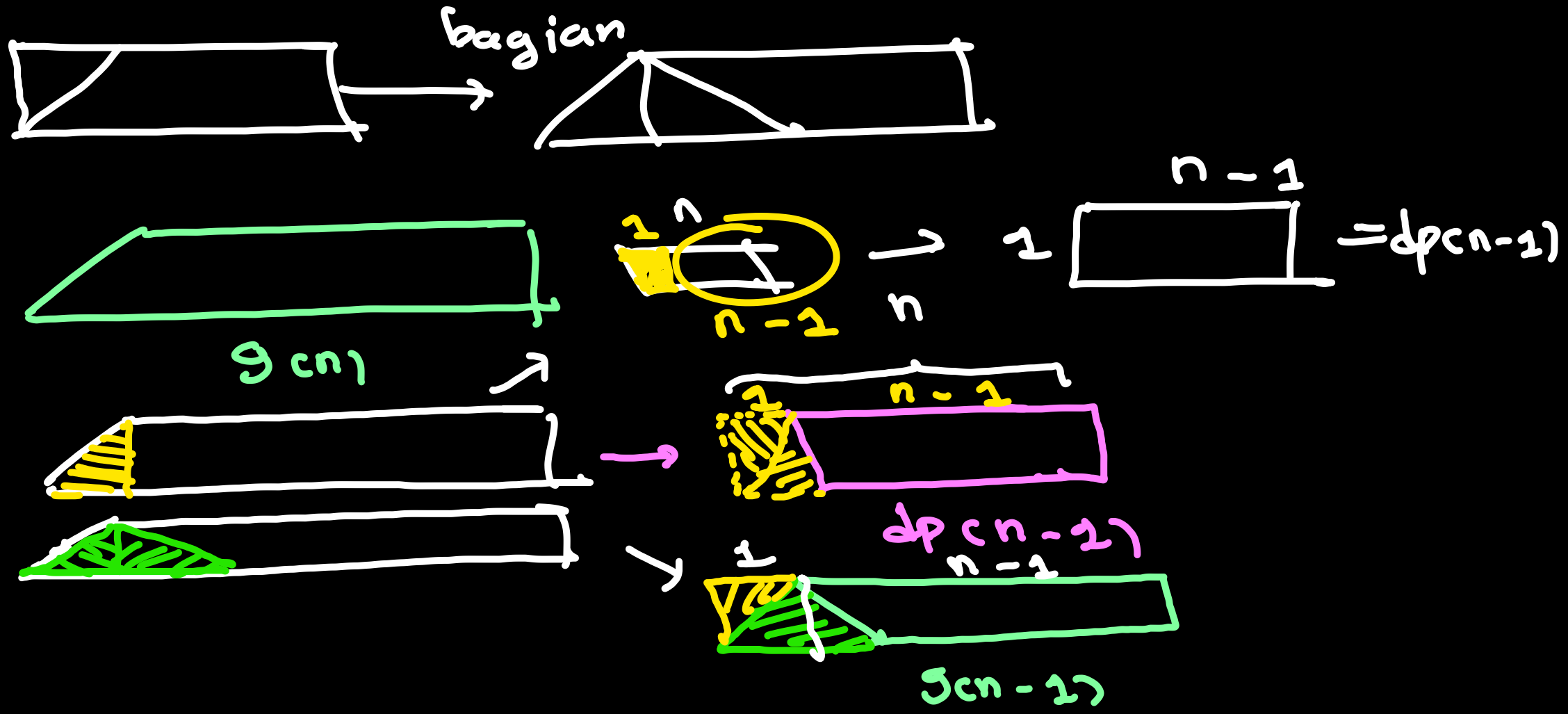
dp(n) = menyatakan banyaknya cara memasang karpet berukuran 1 x n



g(n) = banyaknya cara memasang karpet pada trapesium berukuran 1 x n



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



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

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

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

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

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

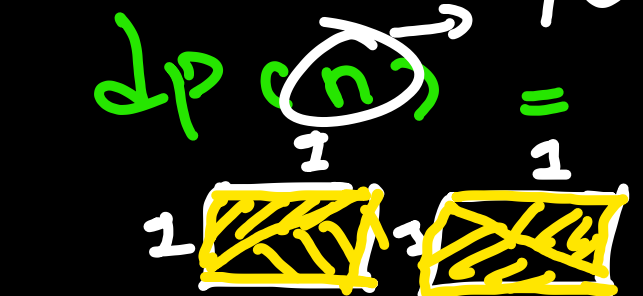
$g(n) = 3g(n-1)$

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

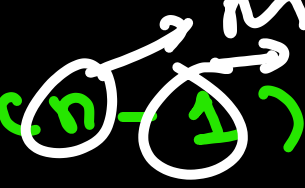
$= 6g(n-1) \rightarrow 3 \cdot 2g(n-1)$

$3 \cdot dp(n-1)$

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



ada 2 cara pasang ke antrik



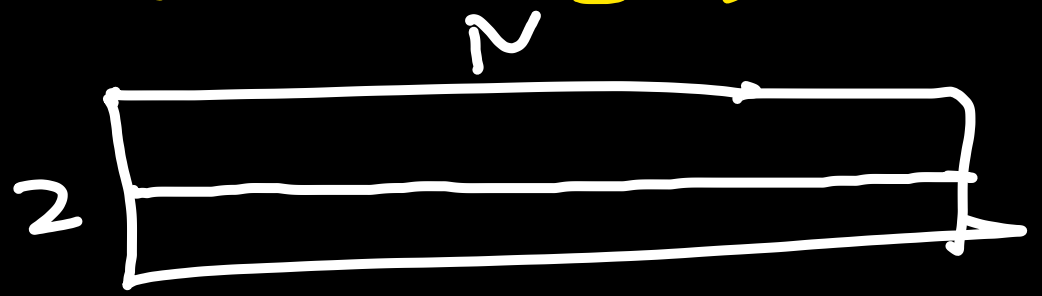
1 base case

$dp(1) = 2$

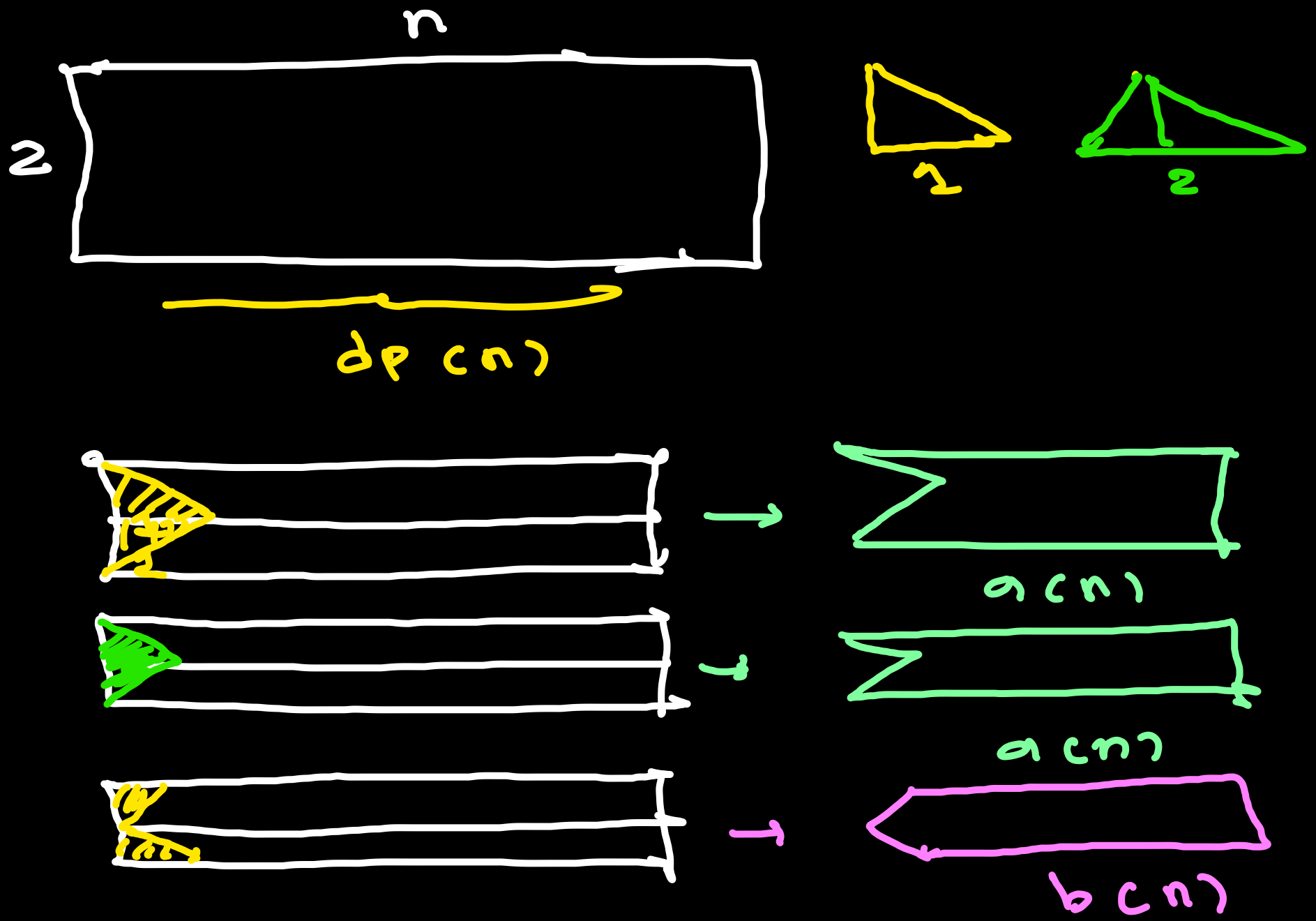
1x1

Subsoal 2 ($1 \leq N \leq 2$)

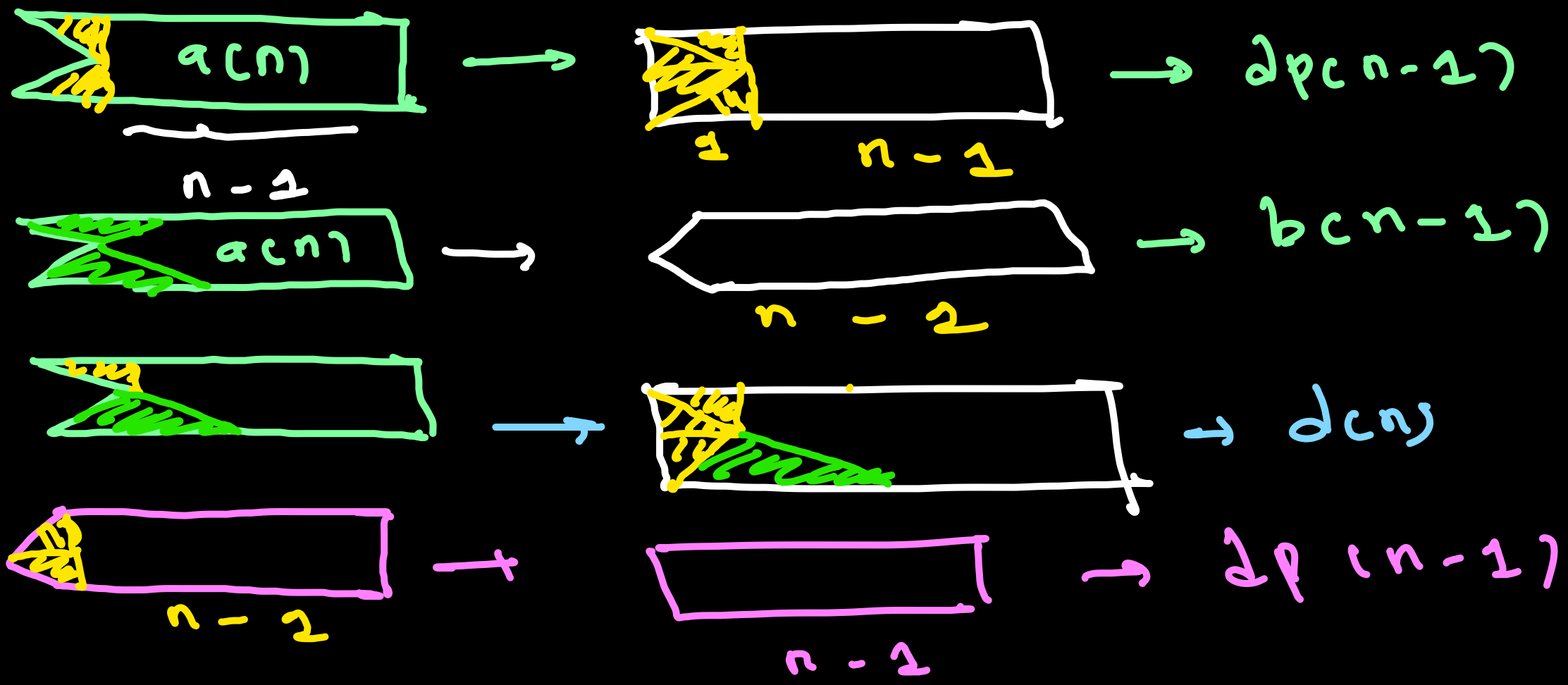
$N = 2$

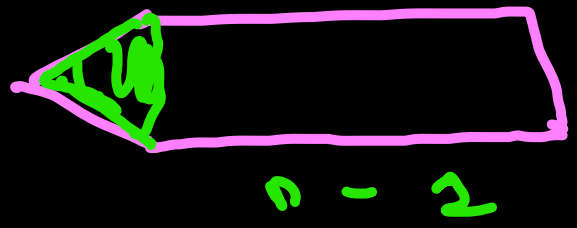


Coba cari dp(n) nya !


n
2
1
2
dp(n)
a(n)
a(n)
b(n)

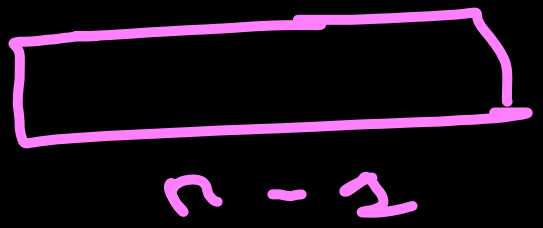
$$dp(n) = 2a(n) + b(n)$$





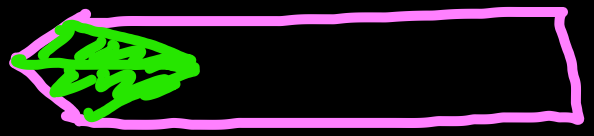
n - 2

→

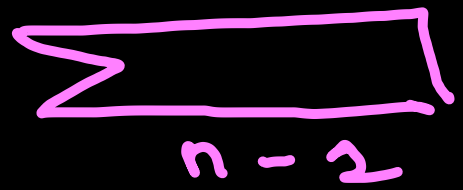


n - 1

→ dp (n - 1)

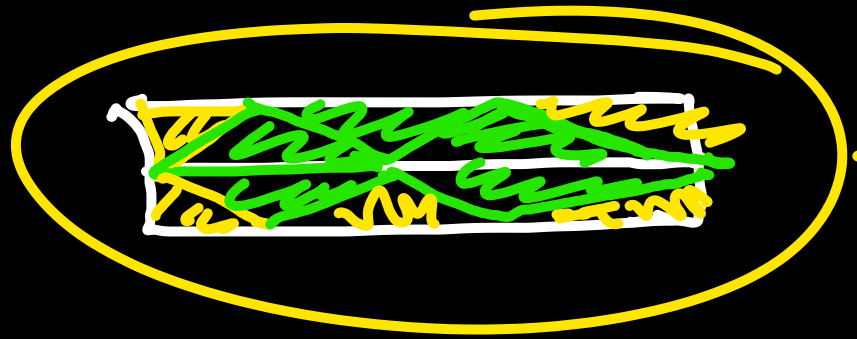


→



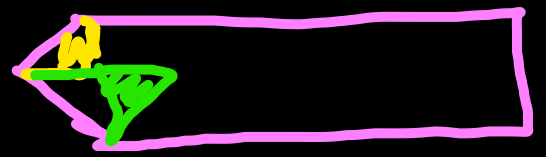
n - 2

→ 9 (n - 1)

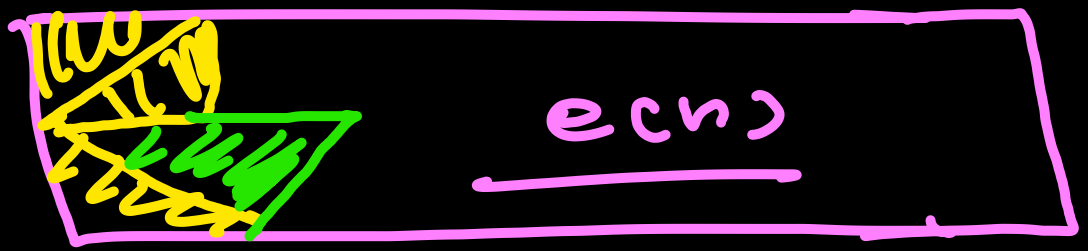


→

bans 1 ≠ bans 2



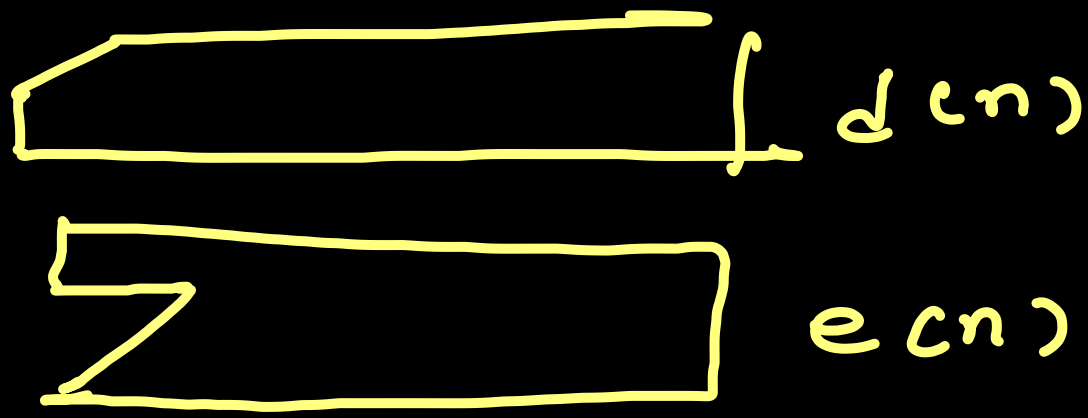
→



e(n)

$$a(n) = d\,p(n-1) + b(n-1) + d(n-1)$$

$$b(n) = 2a\,p(n-1) + a(n-1) + e(n-1)$$



Stack and Queue

- DFS → Recursion / Stack
- BFS → Queue

DFS (Depth First Search)

1 visited

2 visited

dfs(1) → dfs(2)

dfs(3)

3 visited

dfs(4)

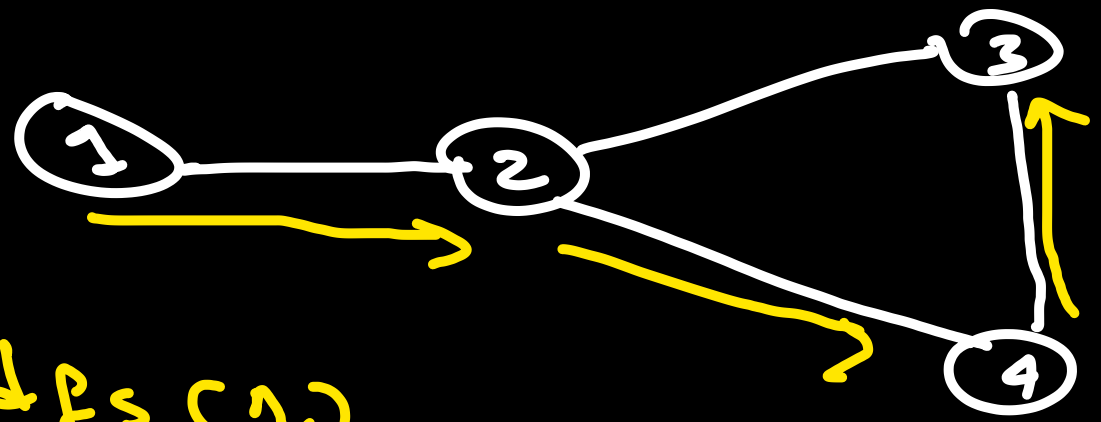
4 visited

dfs(4)

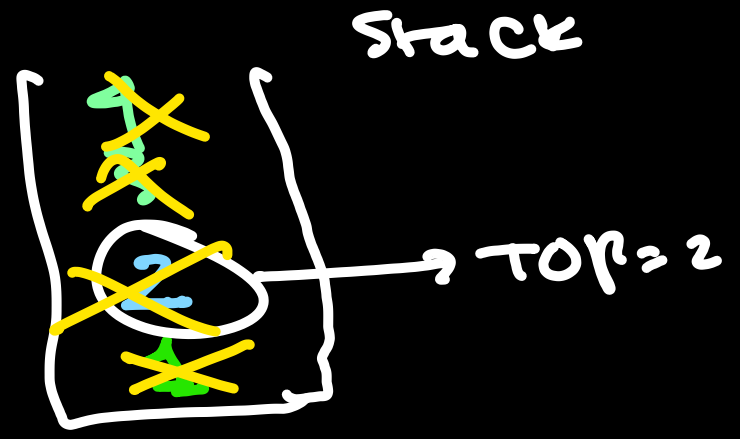
dfs(2)

dfs(3)

```
dfs(n)
→ Mark n visited
dfs(1)
dfs(connection(1))
dfs(2)
dfs(connection(n)):
    (if c(conn(n)=visited)
        continue
    dfs(conn(n))
```



dfs(1)



dfs(2) =

x = 5

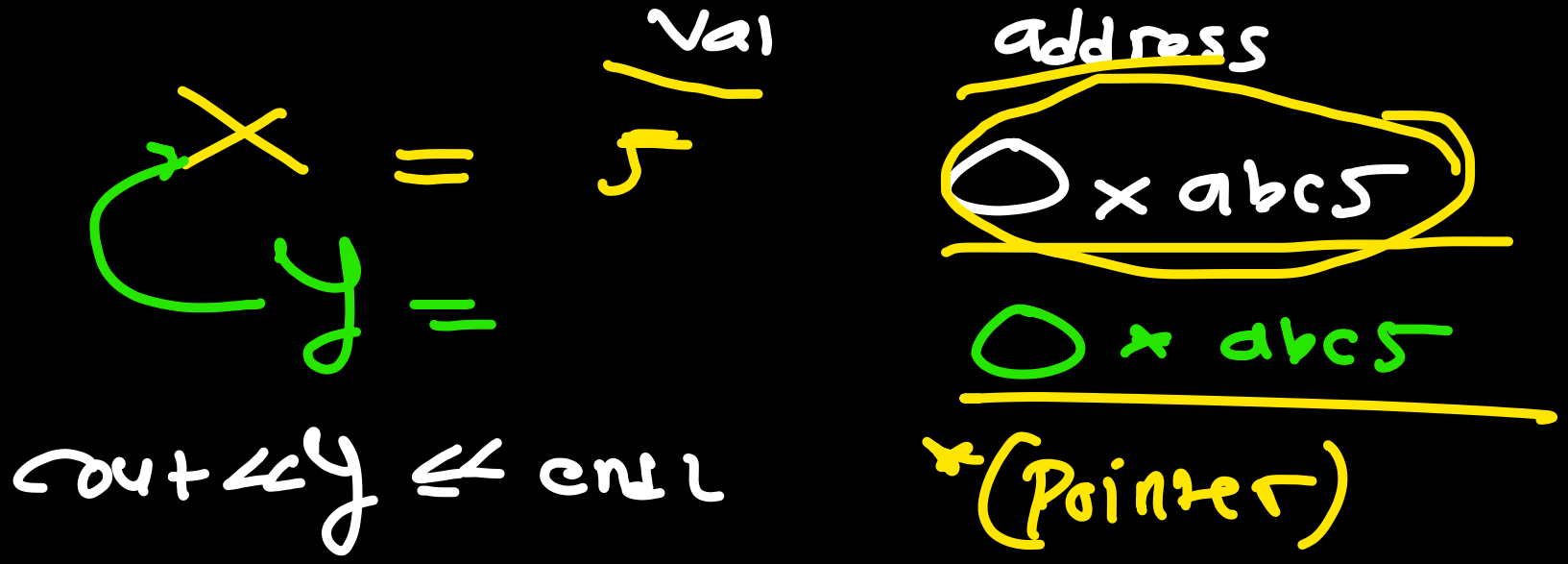
y = x

y = &x

y = duplicate(x)

x = &y

y = y


Val
address
x = 5
0x abc5
y =
0x abc5
cout << y << endl
*(Pointer)
* Var –