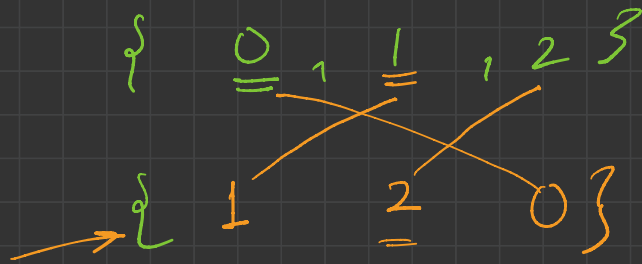
Dynamic Programming

→ i/p →



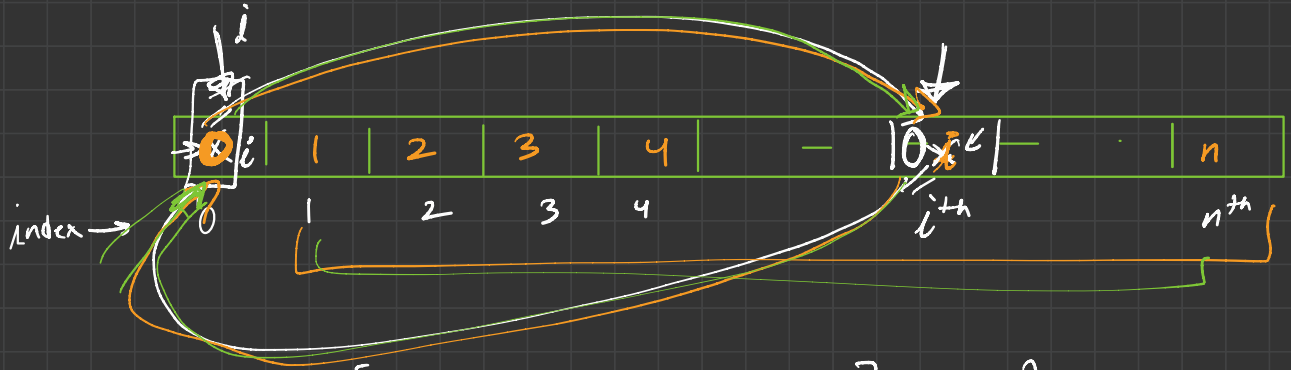
N = 3

{ 0, 1, 2

N = 7

{ 0, 1, 2, 3, 4, 5, 6

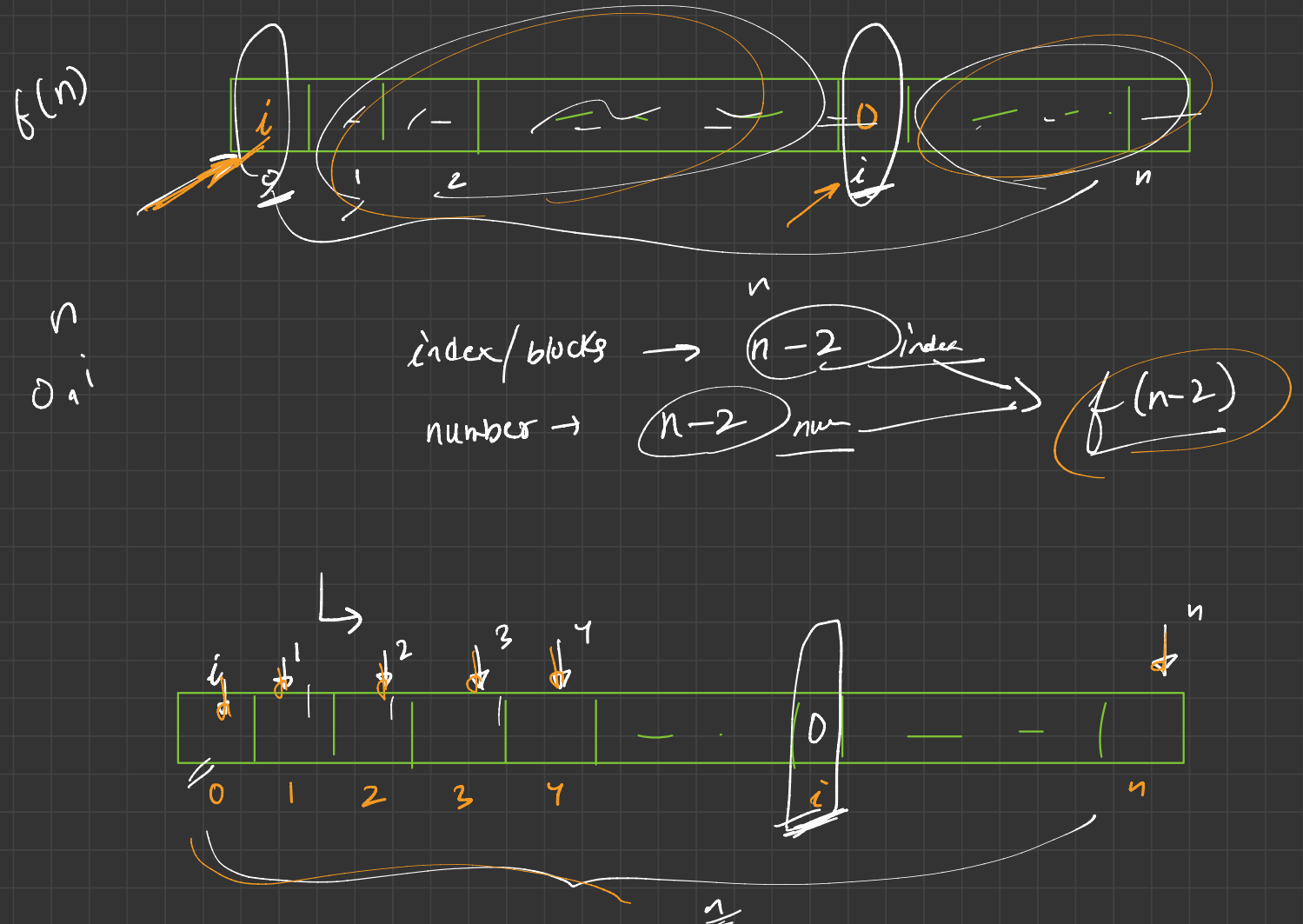
i/p → N → {0, 1, 2, 3, — — — , N-1}



(n-1) * [Solution of Subproblems] → Answer

Two possibilities

→ i & 0 / swapped


f(n)
i
0
1
2
i
n
0 , i
n
index / blocks → n-2 index
number → n-2 num
f(n-2)
i
1
2
3
4
n
0
1
2
3
4
i
n
n

indexs / blocks → n-1

number → n-1 num

⇒ f(n-1)

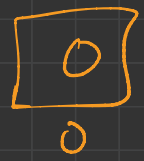
$$f(n) \rightarrow (n-1) * [f(n-2) + f(n-1)$$

N = 2

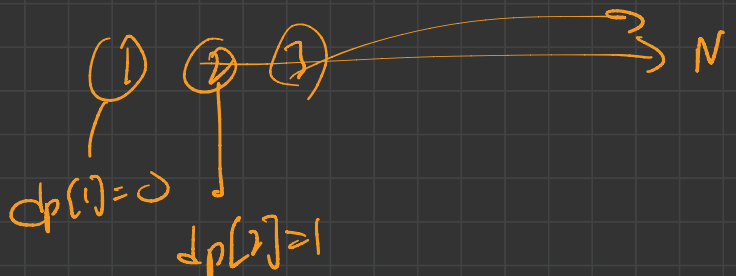
{0, 1}

N = 1

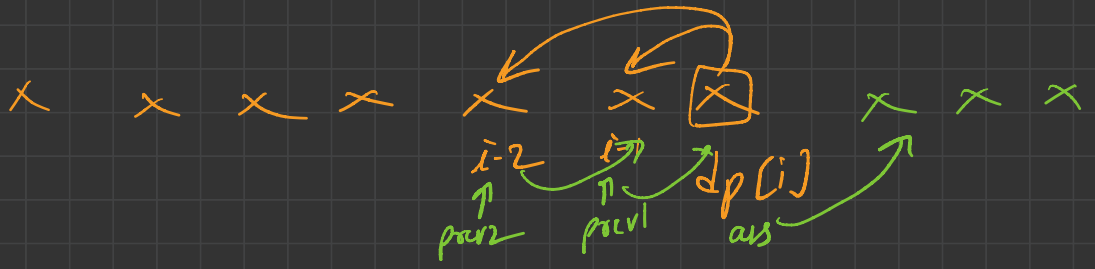
{0}



~~N~~ N>0



5.0 → dp[i] → dp[i-1], dp[i-2]



ans = prev1 + prev2

prev2 = prev1

prev1 = ans

→ O(1)

Rec → expo

Rec + Mem → T.C → O(n)

S.C → O(n) + O(n)

Tab → T.C → O(n)

S.C → O(n)

S.O

↳ S.C → O(1)