

---

---

---

---

---



# Dynamic Programming

i/p  $\rightarrow$  coins  $\rightarrow [1, 2, 5] \rightarrow$  infinite supply  
amount  $\rightarrow [11]$   
minimum no. of coins  $\rightarrow 3$   $[5+5+1]$

coins [2]

target  $\rightarrow 3$

ans  $\rightarrow [-]$

[0]

$$T \rightarrow 0 \xrightarrow{=} 0 \xrightarrow{\text{if}} 0 \text{ coin}$$

[1, 2, 3]

$\times \rightarrow 7$

$$(1+1+1+1+1+1+1) \rightarrow 7$$

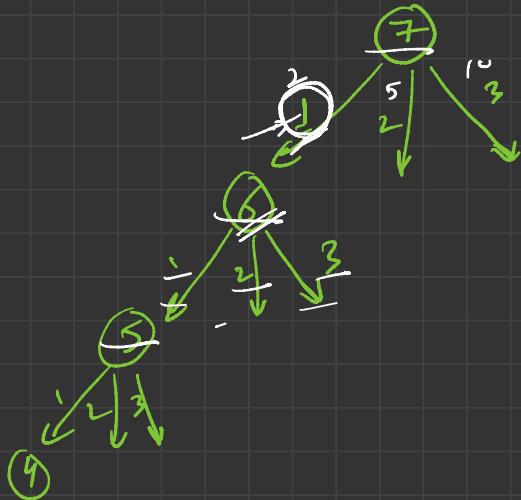
$$2+2+2+1 \rightarrow 4$$

3+3+1

3

→ Rec' + Mem' → tabulation → S-O

# ① Recursion



[1, 2, 3]

7  
[2, 1, 0]

// B.C

if (target == 0)  
return 0;

if (target < 0)  
INT\_MAX

int min = INT\_MAX;

for (no of coins)  
int ans = f(amt - coin[i]);

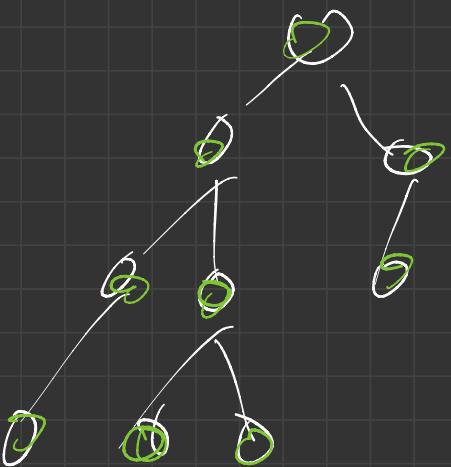
memoization

$\text{if } (\text{ans}! = \text{INFINITY})$

$\text{mini} = \text{mini} (\min_i \prod_{j=1}^n t_{i,j})$

3

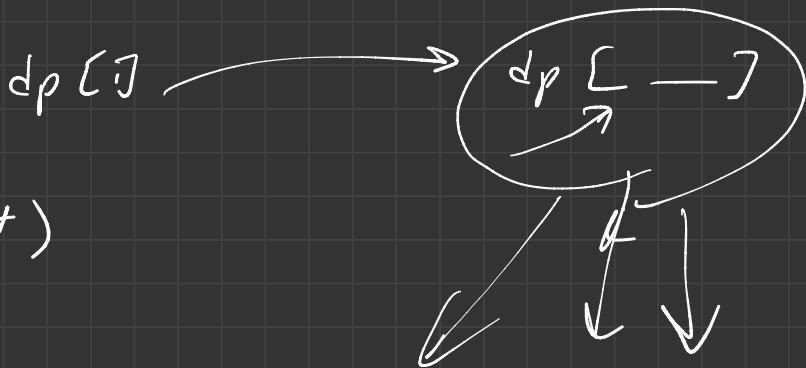
other ans



(dp[i])

(n)

$\Rightarrow (k) \rightarrow \text{min no of coins}$



for ( $i \rightarrow \leq \text{amt}$ )

$dp[i]$

$dp[i]$

$dp[3]$

$dp[\text{amount}]$

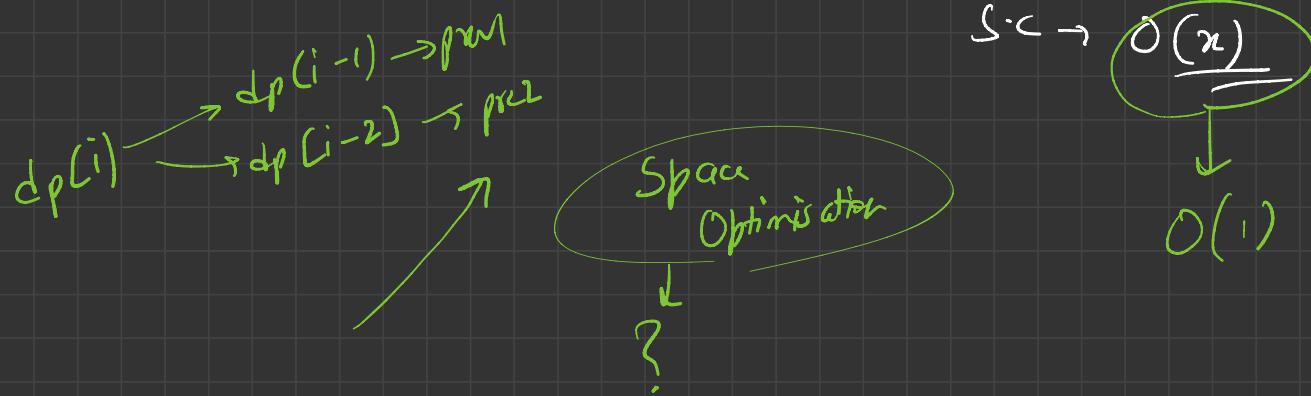
① Recursion :-  $T.C \rightarrow \underline{\text{CPO}}$

$T.C \rightarrow \underline{n^*n}$

amount no of coins

$S \leftarrow O(x)$

② Tabulation  $\rightarrow T.C \rightarrow \underline{n^*n} \{$



$S.C \rightarrow \underline{\underline{O(n)}}$   
 $\downarrow$   
 $O(1)$

$dp[i] \rightarrow dp[i - \underbrace{\text{num}[j]}_l];$

$i-1$

$i-2$

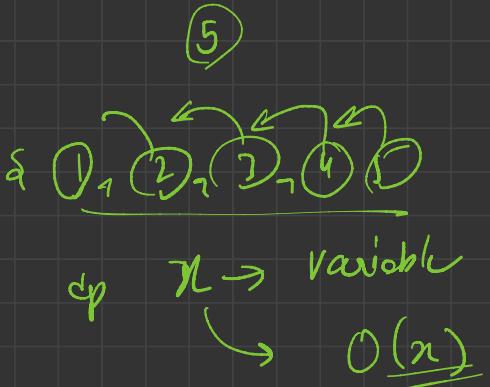
$i-3$

$i-4$

$i-5$

$i$

$i+1$



$\downarrow p[0]$   $\downarrow p[1]$   $\downarrow p[2]$   $\downarrow p[3]$   $\downarrow p[4]$  - - -  $\downarrow p[n]$

A diagram illustrating a sequence of memory locations. Five circles, each containing a value from 0 to 4, are arranged horizontally. A green arrow originates from the circle containing '1' and points to the circle containing 'n', spanning all the intermediate circles.

























