Dynamic Programming Class-3

Special class

Love Babbar • Jan 10, 2024

→ Painting fence

n → post, K → colors

Condⁿ → no two adjacent posts have the same color

n = 1
K = 3 → R, G, B

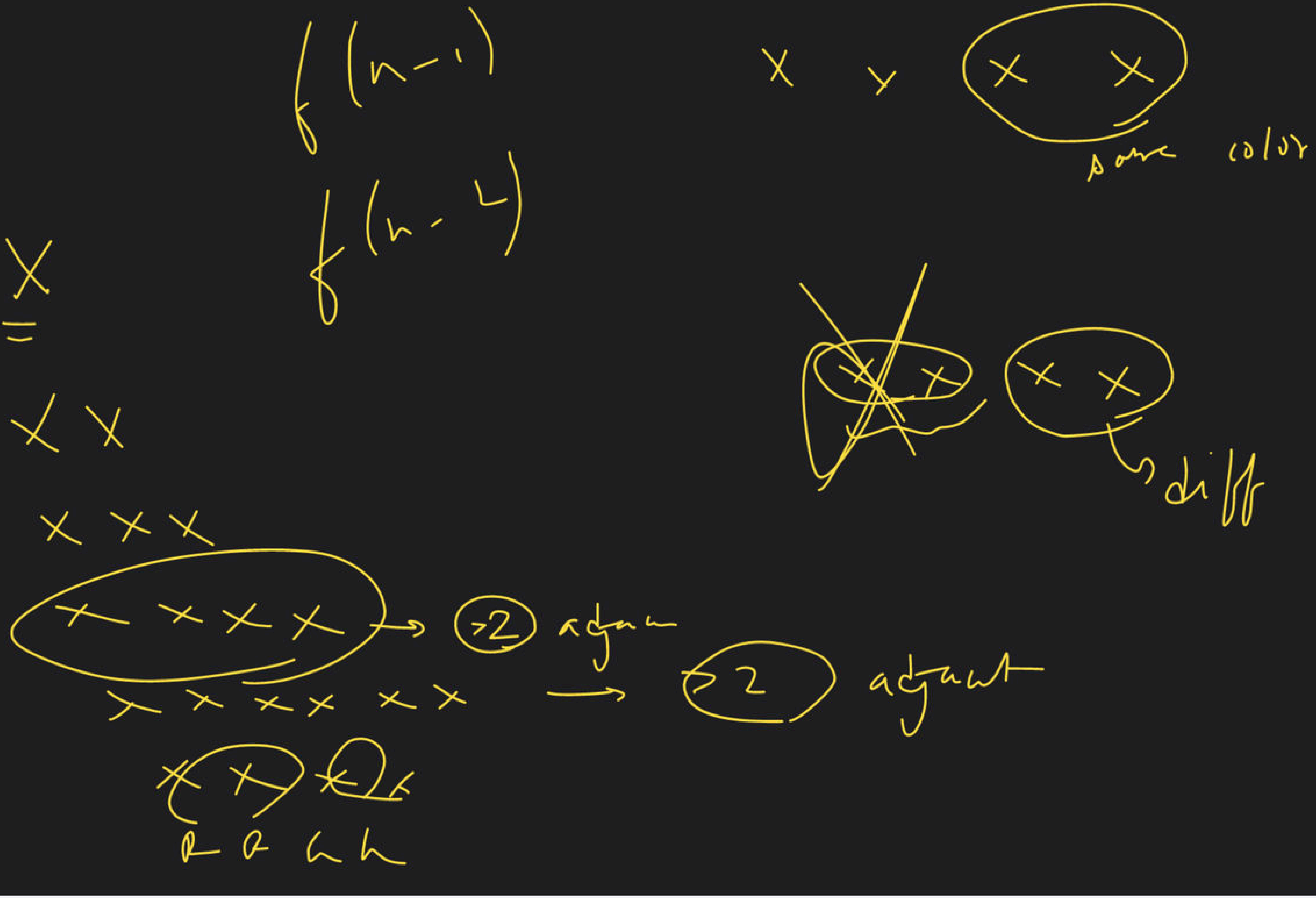
R || G || B → ans = 3

n = 2 →

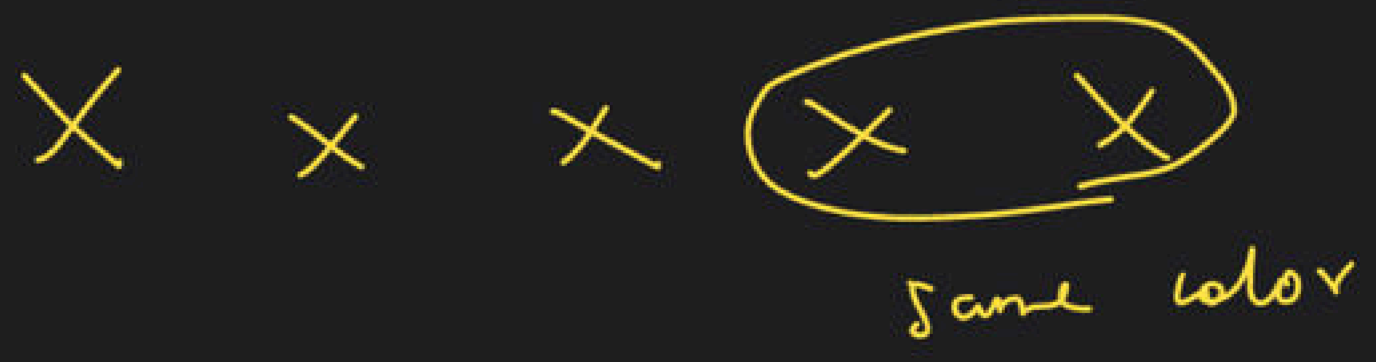
RR GG BB
RG GR BR
RB GB BG

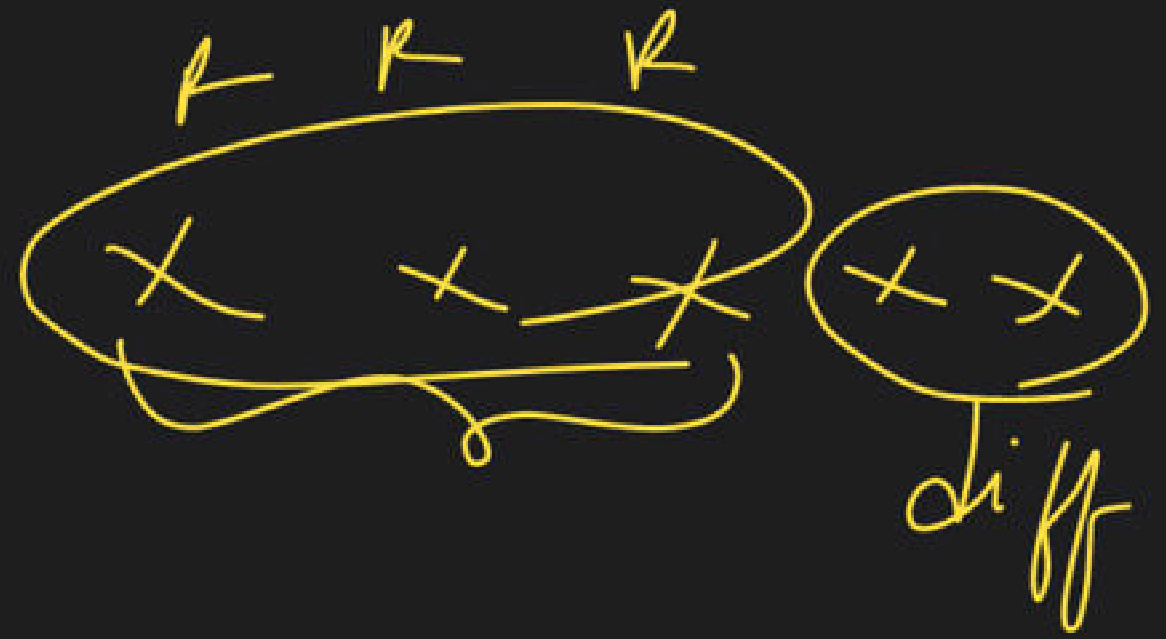
→ ans = 9

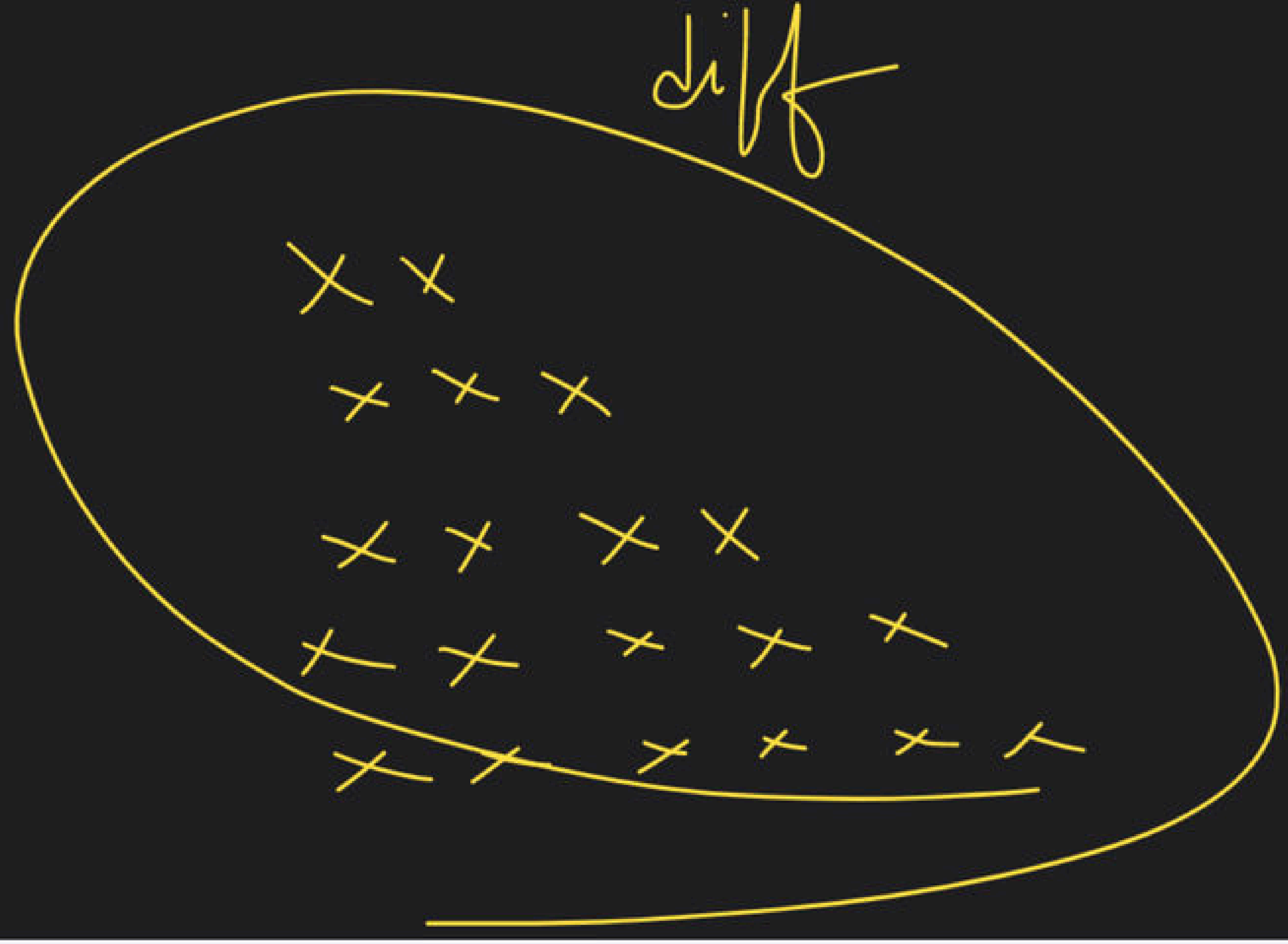
n = 3 →


f(n-1)
f(n-2)
same color
diff
>2
adjacent
>2
adjacent
R R G G

same







K=3 R G B

R R ⊗ R G B

	n = 1	n = 2	n = 3	n = 4
same	R G B = 3	RR GG BB	R~~B~~B R~~G~~G B~~R~~R B~~G~~G G~~R~~R G~~B~~B	RBGG RBRR RGRR RGBB BRBB BRGG BGRR BGBB GRBB GRGG GBRR GBGG RRBB RRGG GGRR GGBB BBRR BBGG
diff	n=1 ↓ K	RB RG BR BG GR GB	R~~R~~G R~~R~~B G~~G~~R G~~G~~B B~~B~~R B~~B~~G R~~B~~R R~~B~~G R~~G~~R R~~G~~B B~~R~~B B~~R~~G B~~G~~G B~~G~~B G~~R~~G G~~R~~B G~~B~~R G~~B~~G	

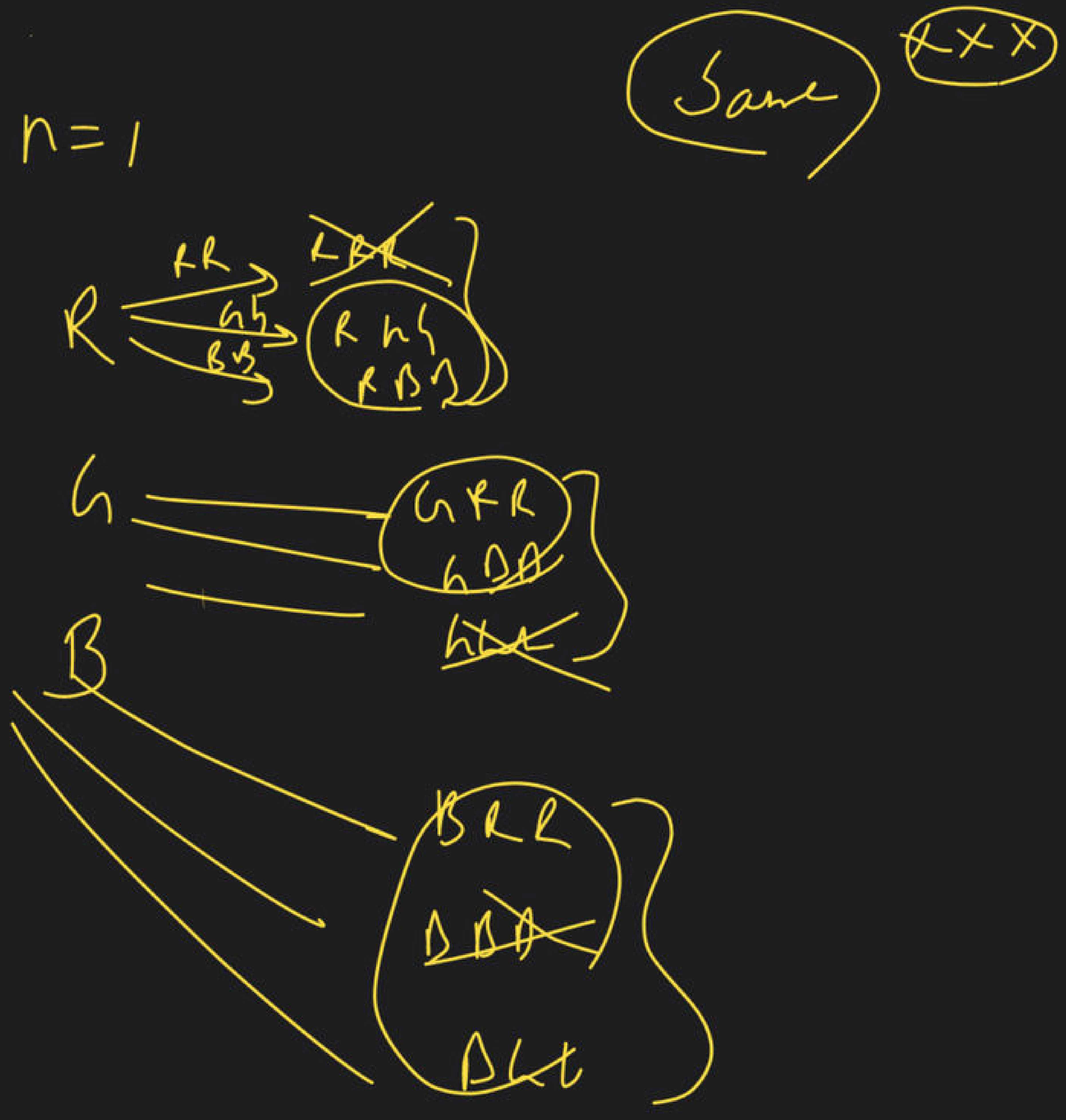
$\Sigma(2) * (K-1)$

α

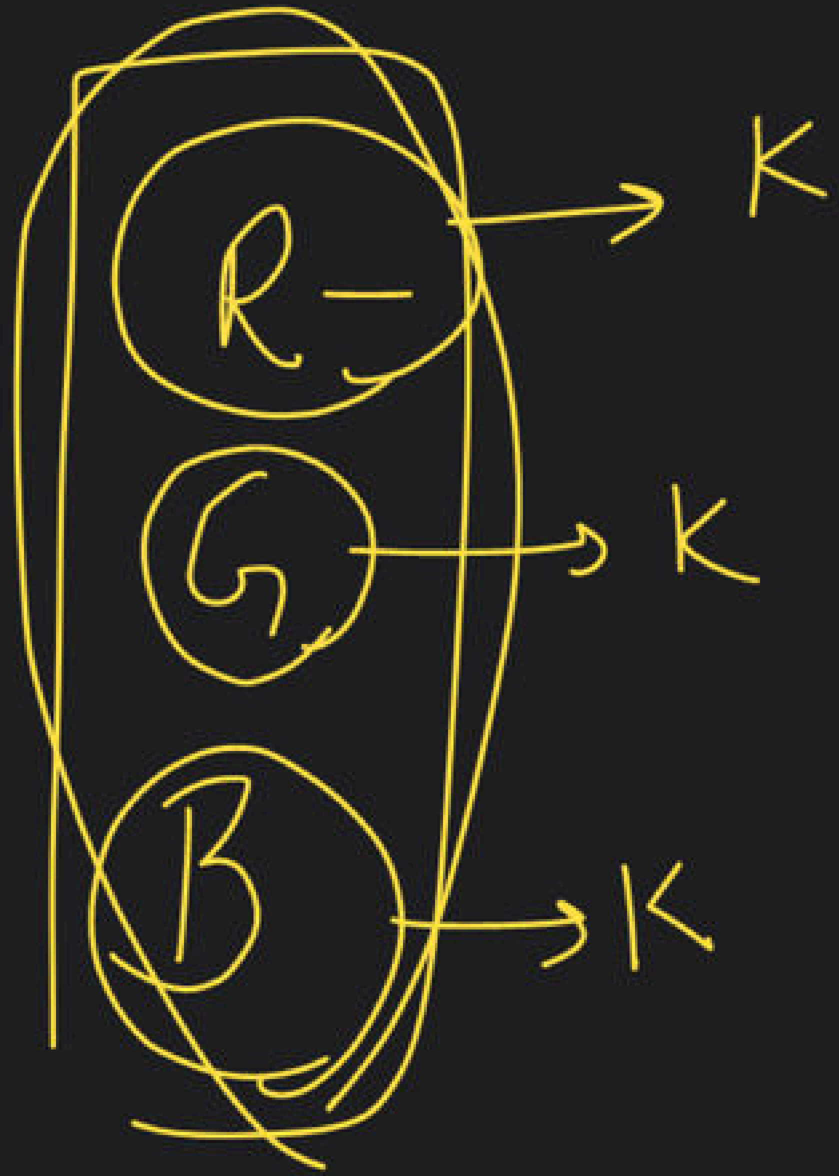
18

9

6 × 2

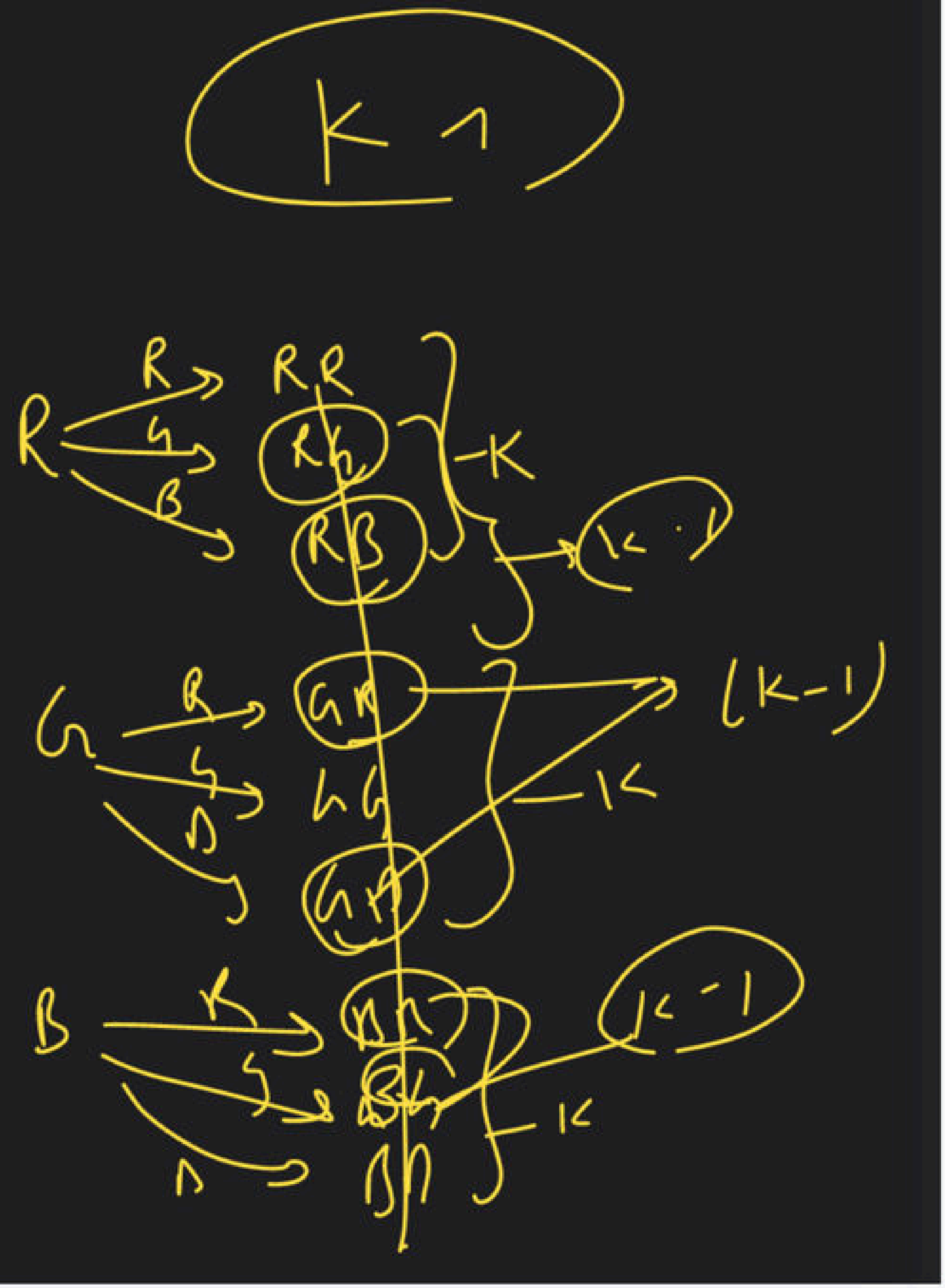

Same
XXX
n=1
R
RR
GG
BB
RRR
RGG
RBB
G
GRR
GBB
GGG
B
BRR
BBB
BGG

Single post



R
G
B
K
K
K

diff



K 1
R
R
G
B
RR
RG
RB
K
K-1
G
R
G
B
GR
GG
GB
(K-1)
K
B
R
G
B
BR
BG
BB
K-1
K

$$f(n) = \text{same}_n + \underline{\text{diff}_n}$$

$$= [f(n-2)](k-1) + \underline{[f(n-1)]}(k-1)$$

$$\boxed{f(n) = (k-1)\left[f(n-1) + f(n-2)\right]}$$

X X

same → RR
GG
BB

X X

diff RG
RB
BR
BG
GR
GB

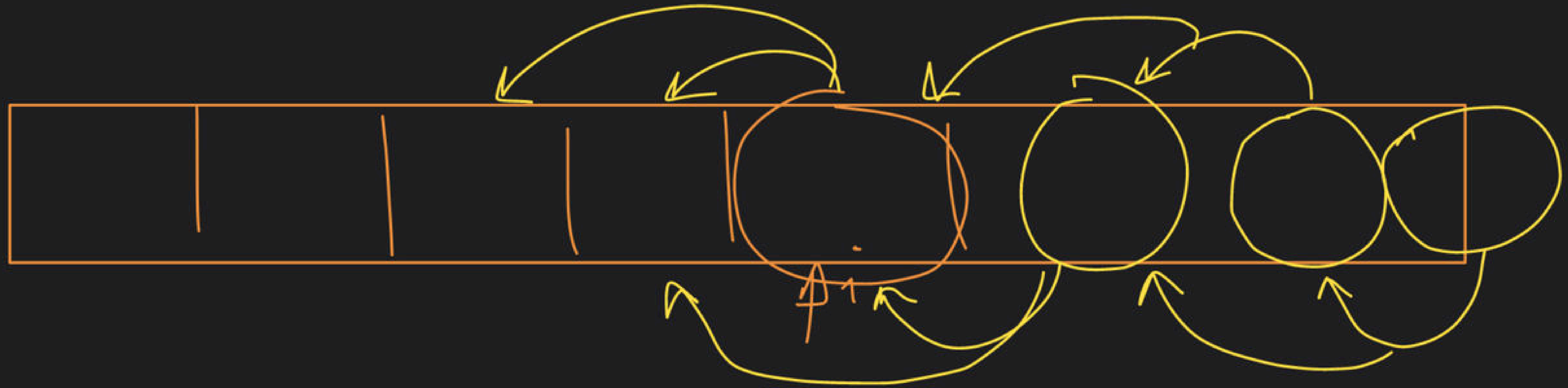
X X X

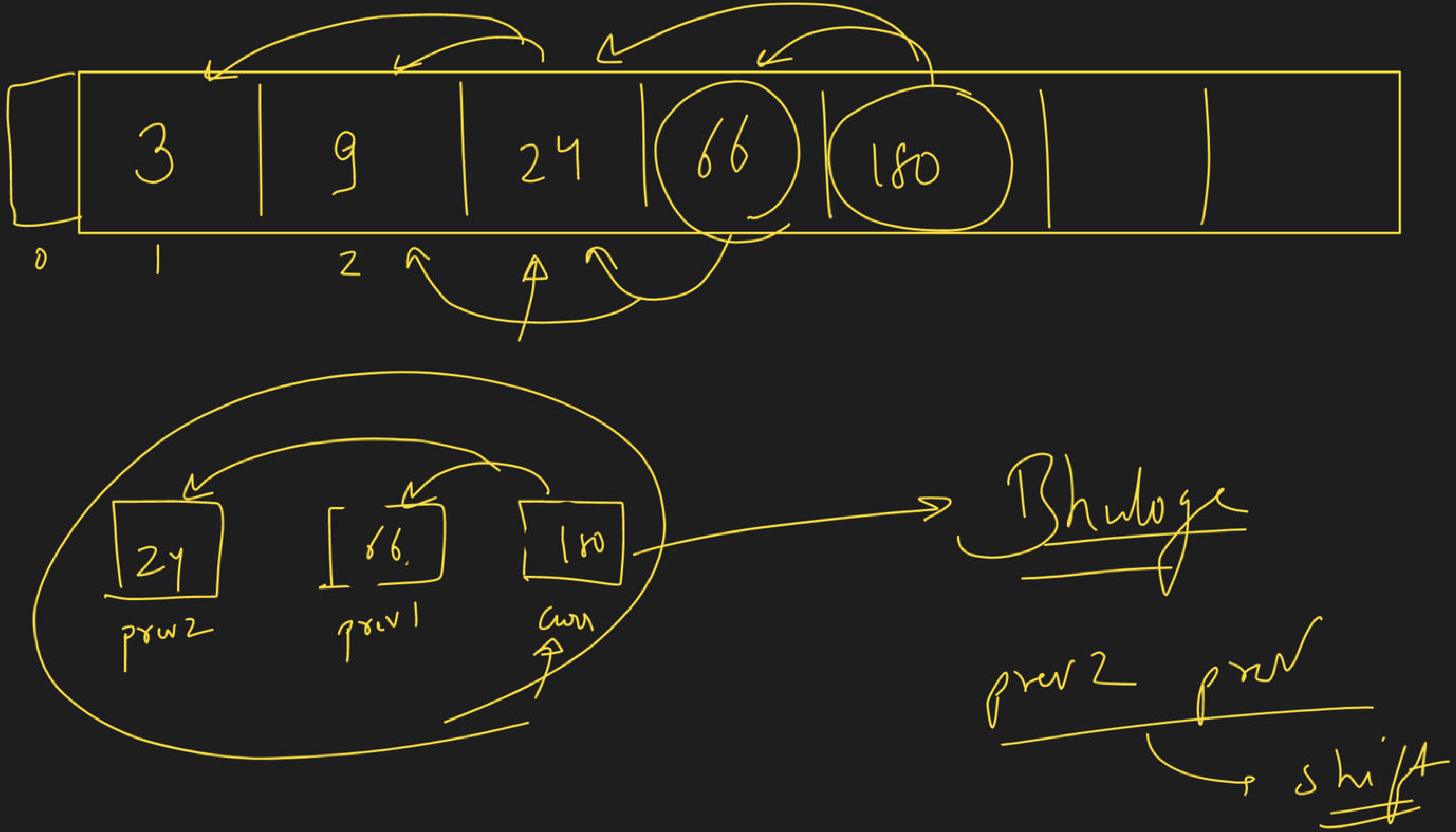
$K = 3$

	$n = 1$	$n = 2$	
same	R G B	RR GG BB $\boxed{K}$	$n = 2$ ~~ans = K + K(K~~ $\boxed{ans = K + K(K - 1)}$
diff	ans = $\boxed{K}$	XX RB RG GR GB BR BG $K(K - 1)$	

Space Optimisation

$$\begin{tikzcd}
dp[i] \arrow[r] \arrow[dr] & dp[i-1] \\
 & dp[i-2]
\end{tikzcd}$$

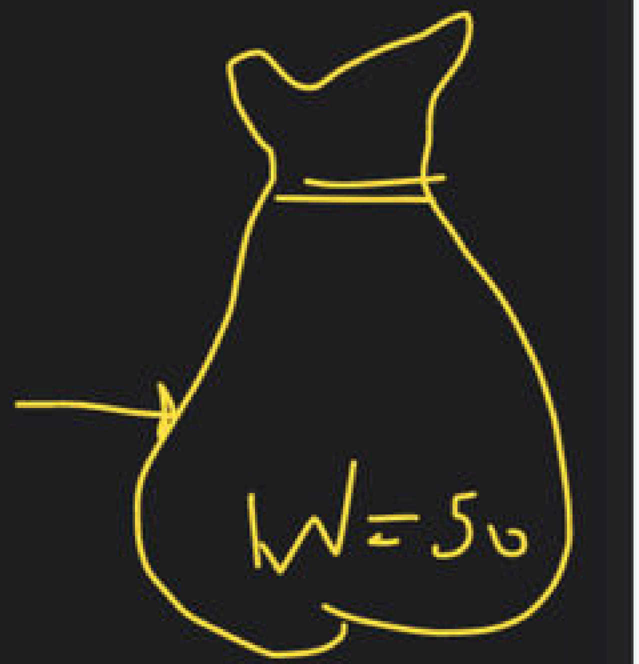



3
9
24
66
180
0
1
2
24
66.
180
prew2
prev1
curr
Bhuloge
prev2 prev
shift

0/1 Knapsack Problem

capacity

W



item insert

maximum profit

n = 3

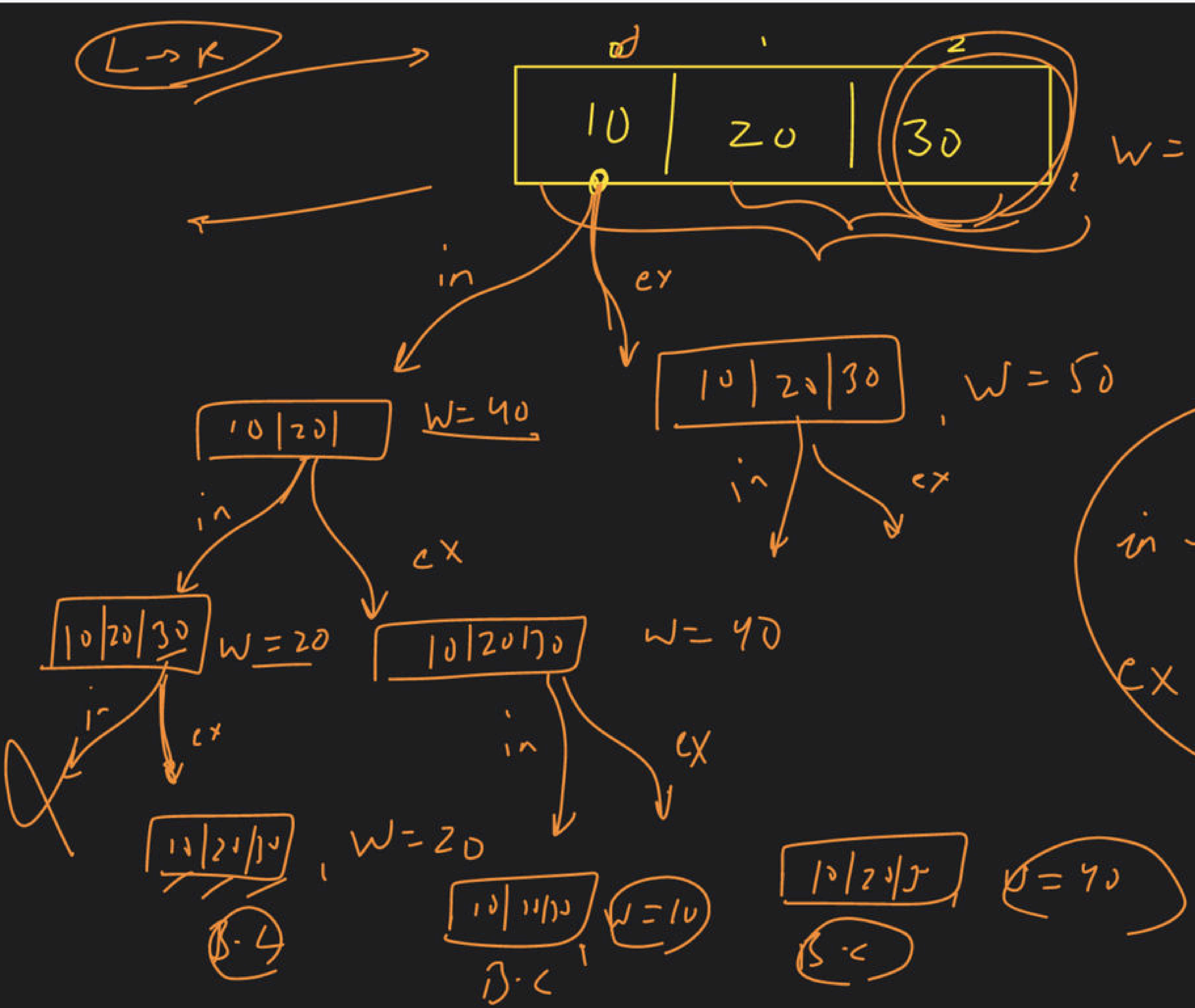
220

weight

10	20	30
0	1	2

Value/profit

60	100	120
0	1	2



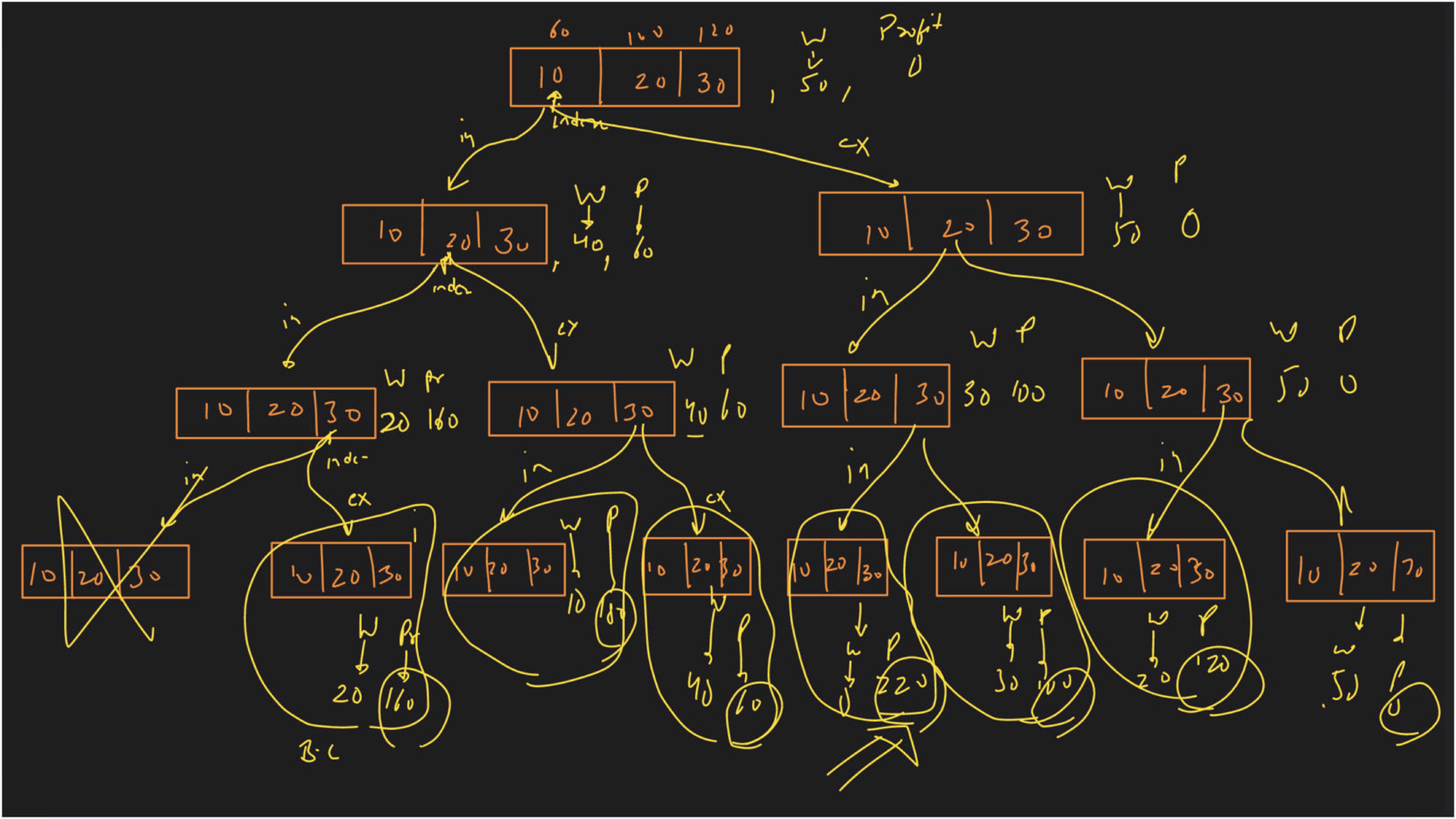
// B.Case

```
if (Index == n-1)
{
    if(wt[index] <= kepi
    { include
    }
```

in → curr item + profit + Rec

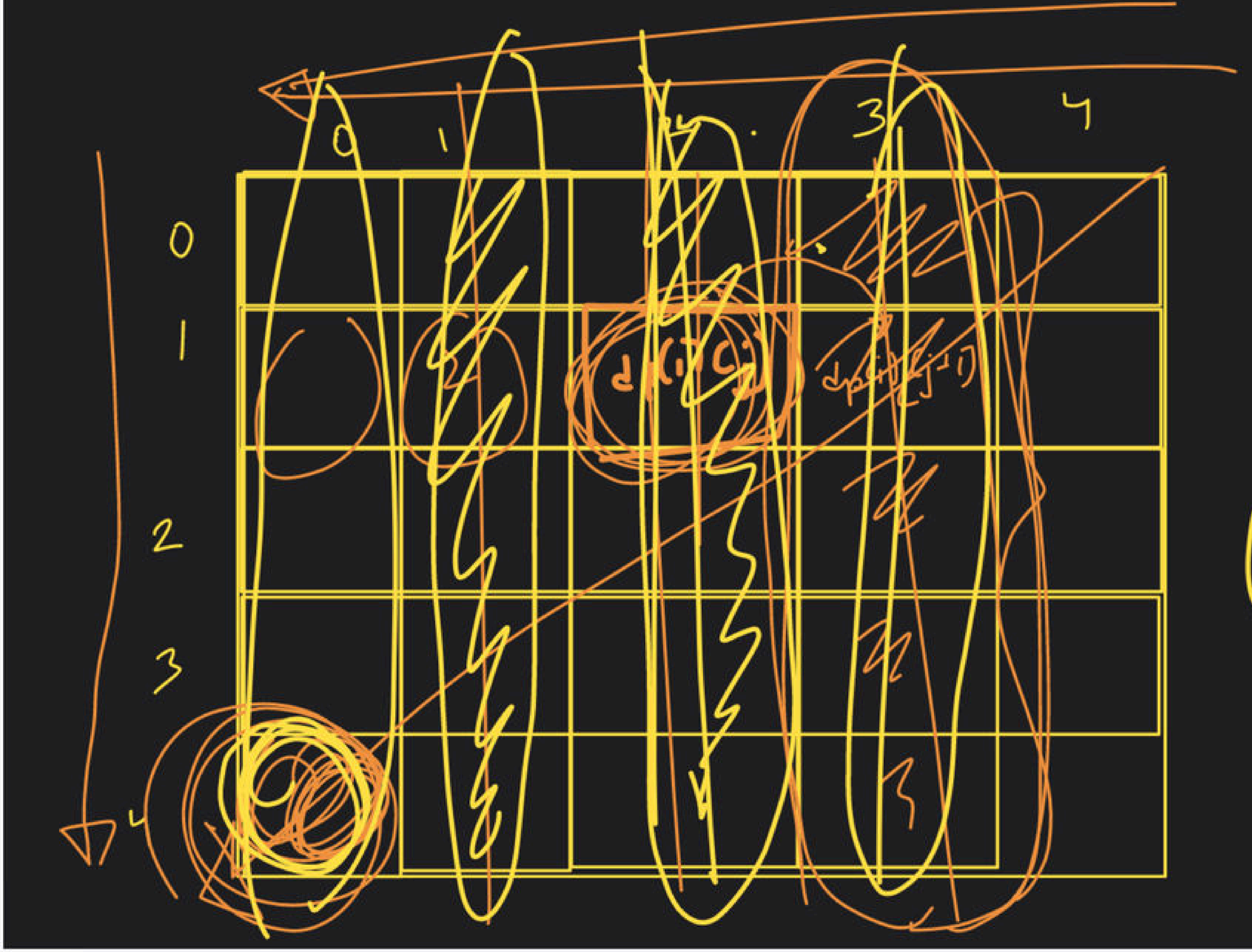
ex → 0 + Rec

max → return ans


60 100 120
10 | 20 | 30
W 50
Profit 0
index
in
cx
10 | 20 | 30
W P
40 60
index
10 | 20 | 30
W P
50 0
in
cx
10 | 20 | 30
W Pr
20 160
index
10 | 20 | 30
W P
40 60
in
10 | 20 | 30
30 100
W P
10 | 20 | 30
W P
50 0
in
10 | 20 | 30
in
cx
10 | 20 | 30
W Pr
20 160
B.C
in
10 | 20 | 30
W P
10 180
cx
10 | 20 | 30
W P
40 60
in
10 | 20 | 30
W P
220
10 | 20 | 30
W P
30 100
in
10 | 20 | 30
W P
20 120
10 | 20 | 30
W P
50 0

i = 1 j = 2

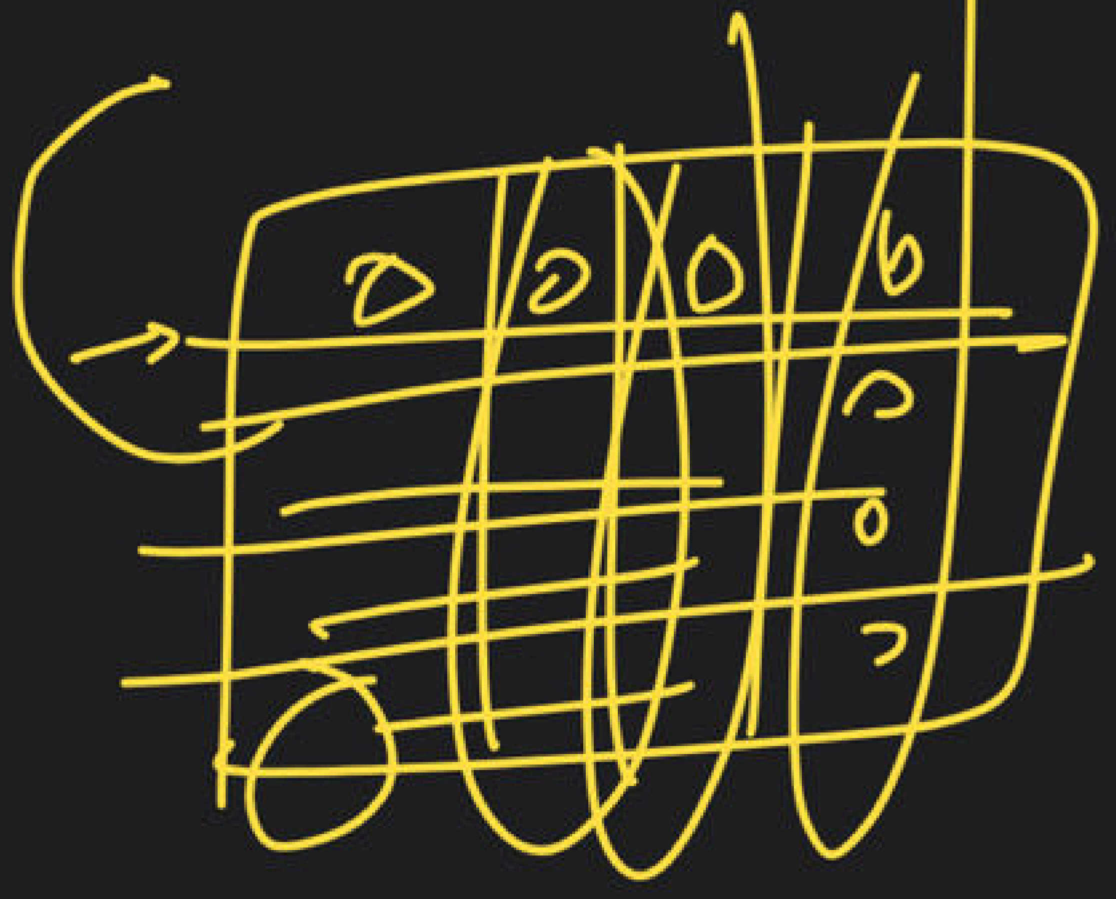
for



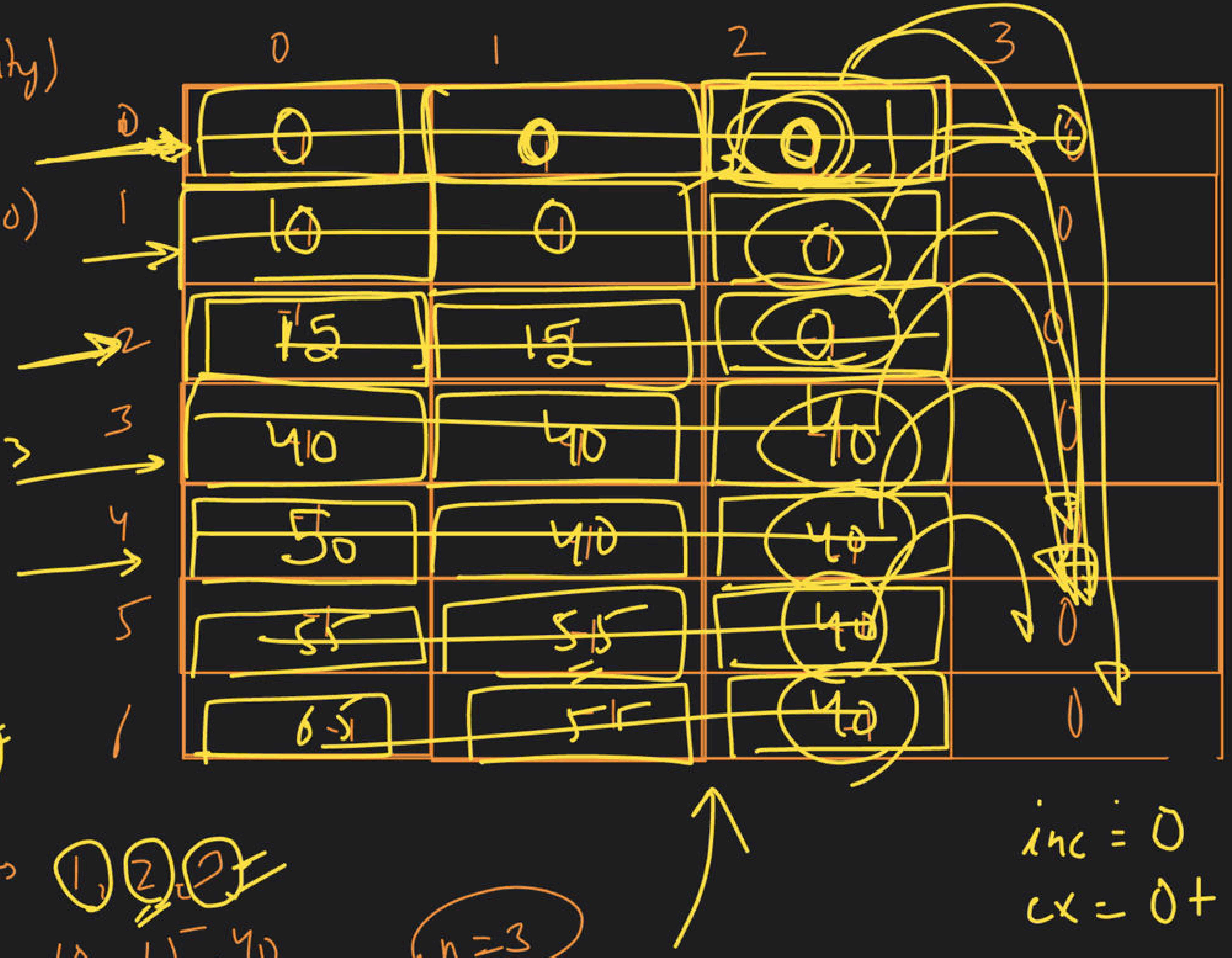
$$\begin{tikzcd} dp[i][j] \arrow[r] \arrow[d, bend right] & dp[i][j+1] \\ & dp[i-wt[j]][j+1] \end{tikzcd}$$

$i-wt[j]$ → koi bhi row ho skta h

$[j+1]$ → next col



```
for( u → <= capacity)
{ for( n-1 → >=0)
```



	0	1	2	3
0	0	0	0	0
1	10	0	0	0
2	15	15	0	0
3	40	40	40	0
4	50	40	40	0
5	55	55	40	0
1	65	55	40	0

n = 3
n - 1 = 2

Capacity = 0

$\boxed{\begin{array}{l} \text{capacity} = 0 \\ \text{indent} = 2 \end{array}}$

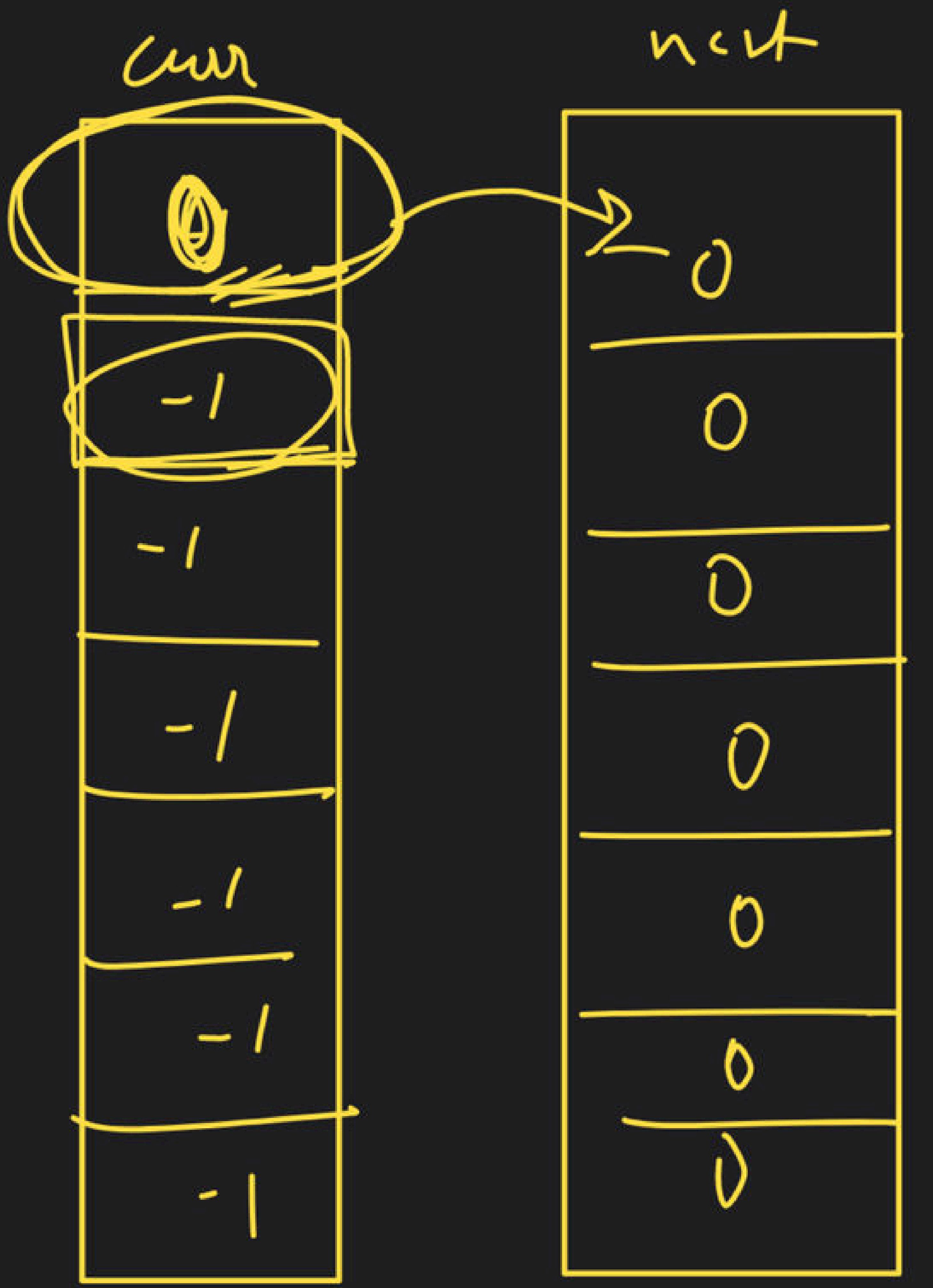
Capa → 1

wt → 1, 2, 2

pr → 10, 15, 40

n = 3

inc = 0
ex = 0 +



i=0

0 -

wt = 2
capa = 0
→ i=0
ex → 0

$$dp[i][j] \longrightarrow dp[i][j+1]$$

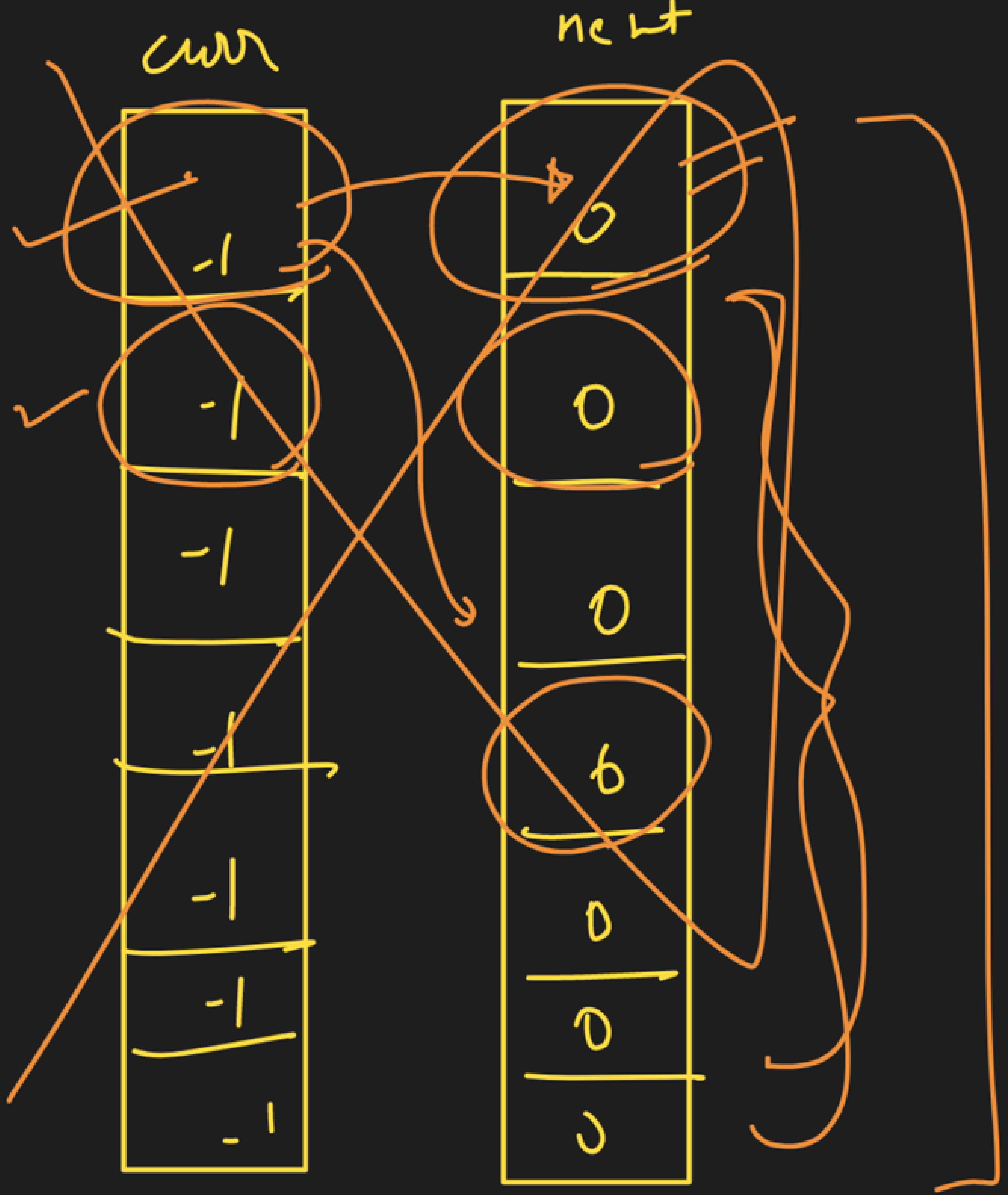
$$dp[i][j] \longrightarrow \underline{dp[i-wt[j]][j+1]}$$

curr	next
0	0
-1	0
-1	0
-1	0
-1	0
-1	0
-1	0

capacity → 0

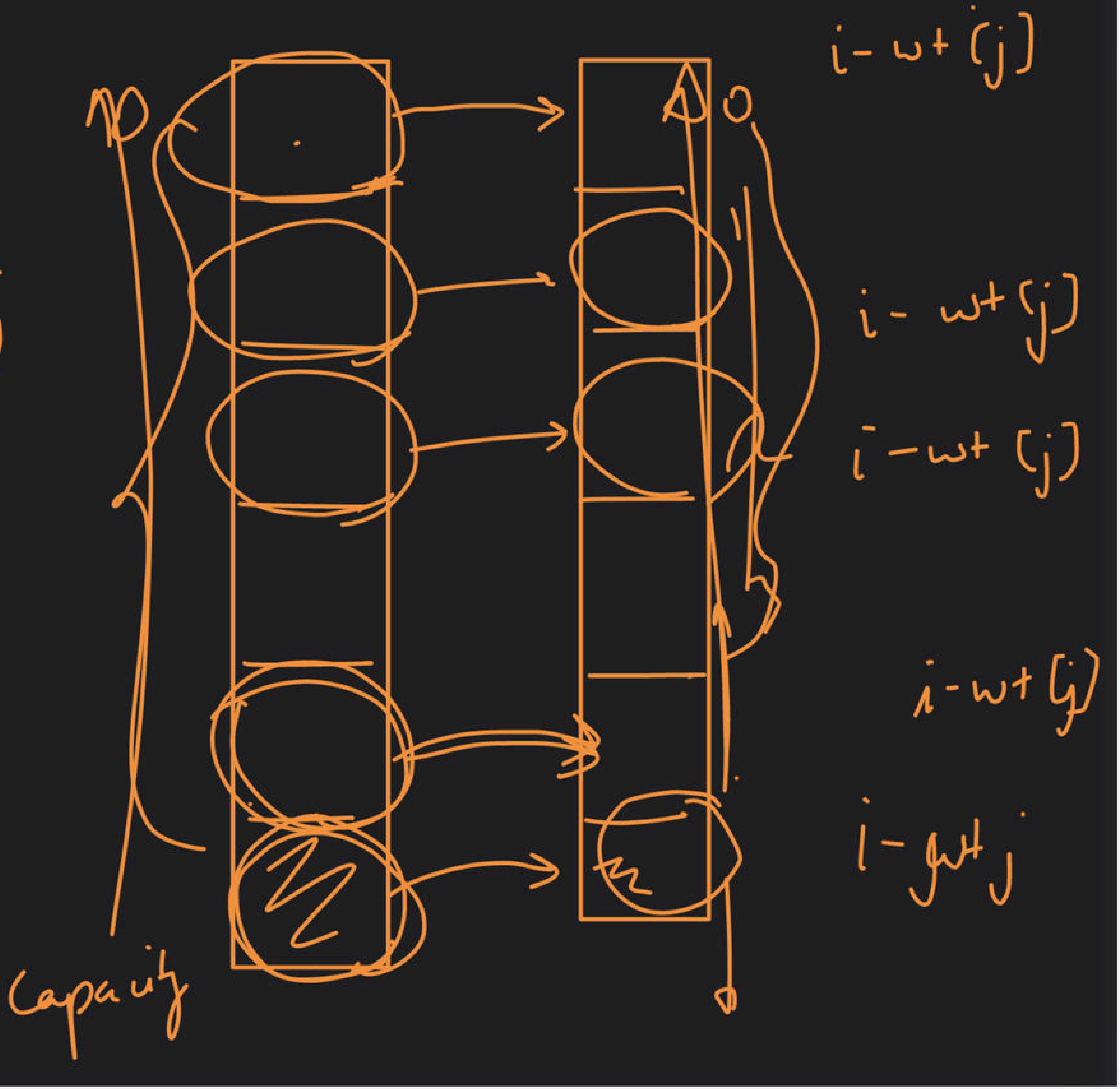
j → n-1 → 0

1	2	3
	↑	↑


curr
next
-1
-1
-1
-1
-1
-1
-1
0
0
0
6
0
0
0

```
        col
for (n-1 ——→ 0) → j
{
          row
    for (0 ——→ capacity) → i
    {

    }
}
```



0
capacity
0
i-wt[j]
i-wt[j]
i-wt[j]
i-wt[j]
i-wt j
0