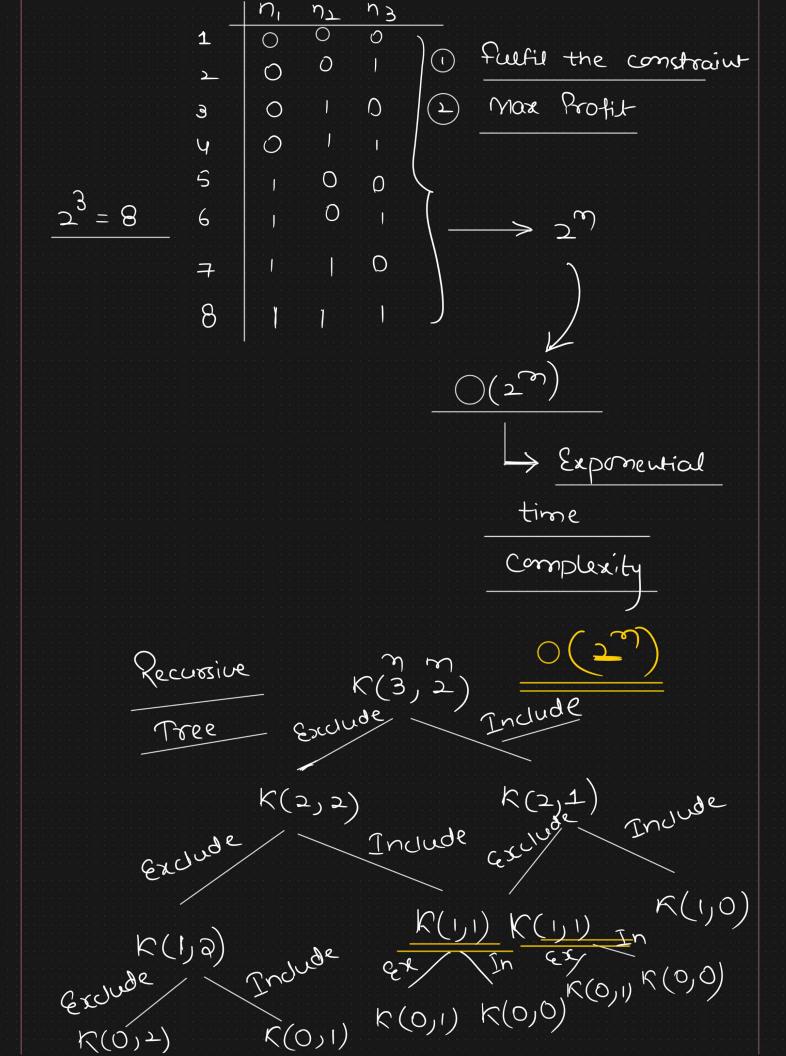


Objects	1	2_	3	
90011	10	٥د	30	
Weight	1	1	1	
X;	9/1	0/1	9/1	
(Binary)				
	m-			

Maximum Profit



Recursine

$$k(m, \omega_t, p_{\delta}, m)$$
 of Base case if $(n = -0)$ d return 0',

Recursive function call

if (weighth-1) >m) of

1) Exclude that Object

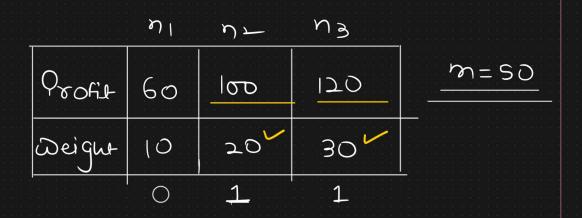
 σ etum $R(m, \omega_t, p_0, n_{-1});$

eue d

return max (K(m, wt, pr, n-1),

90(n-1) + K(m-Wt(n-1), Wt)

pr, m-1));



$$\frac{dp(n+1)(m+1) \rightarrow 2D Array}{columns}$$

to store all the unique function call values

Quoid the

Problem of

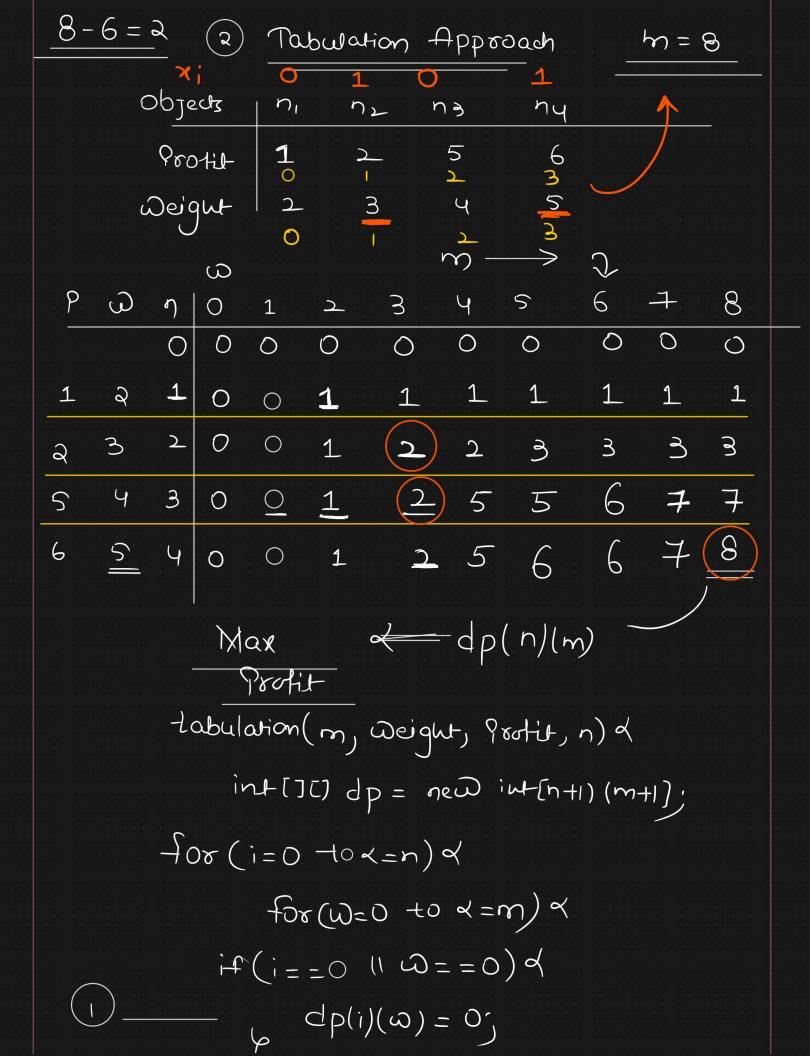
Re-computation

time complexity

(n * m)

Space complexity: O(n+m)

7 No Recursion



$$if(\omega eight(i-1) x = \omega) x$$

dp(i)(w) = max(dp(i-1)(w), profit(i-1)+

dp(i-1)(w- weigne(i-1)),

p

time complexity > mxm

Space complexity nxm