You have n objects with weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$.
You have a sack which can carry atmost wt. W.
↳ $P(W, w[1..n], v[1..n])$

The n^{th} object may or may not be present:

n^{th} object not present : $P(W, w[1..n-1], v[1..n-1])$

n^{th} object present : $P(W - w[n], w[1..n-1], v[1..n-1])$

Recursion: $dp(W, w[1..n], v[1..n]) \to$ optimal value

$$dp(W, w[1..n], v[1..n]) = \max\left\{ dp(W, w[1..n-1], v[1..n-1]),\ v[n] + dp(W - w[n], w[1..n-1], v[1..n-1]) \right\}$$

Now memoize :

$dp[\text{max weight } W'][k] \to$ if W'-sack allowed then what is the max value I can get out of the first k objects.

weights →

k ↓

k \ weights	0	1	2	3	...
0	0	0	0	0	...
1	0	$\square$			
2	0				
3	0				
⋮	⋮				

Either ignore the first obj $(\to 0)$
or put the put the first object if you can

$w[1] \le 1$

You are given a string. Find the length of a subsequence (not necessarily contiguous) of max length which is palindromic.

→ "abca" ans: "aba"

$dp[i][j]$ = answer for string$[i..j]$

Transition:

$$dp[i][j] =$$

if $(str[i] = str[j])$ then $2 + dp[i+1][j-1]$

else $\max\{dp[i+1][j],\ dp[i][j-1]\}$

abca:
1234

$i \downarrow$, $j \rightarrow$

	1	2	3	4
1	1	1		
2	0	1		
3	0	0	1	
4	0	0	0	1

Given a list L of n non-neg int, and an integer x, determine if there is a subset (multiset) with sum x.

$dp[i][j]$ → Is there a subset of $L[1..i]$ which sums to j? (T/F)

$$dp[i][j] = dp[i-1][j] \underbrace{\text{ OR }}_{\text{logical OR}} dp[i-1][\underbrace{j - L[i]}_{\text{check if } \geq 0}]$$

(if < 0 then just $dp[i-1][j]$)