omsairam omsrivenkatesayanamaha ommahaganadhipatayanamaha omnamahshivayaha

Dynamic Programming

→ Used largely to solve optimisation problems in industries

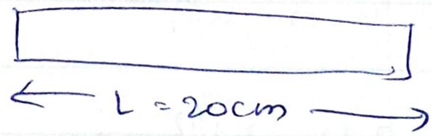
Ex:- Viterbi Algorithm
Junction-tree algorithm
Bellman-ford algo
Dijkstra's Algo

Steps:-(Major)
1) You have to make a sequence of steps
2) Either maximise or minimise

→ Mostly used when we make a "sequence of decisions"

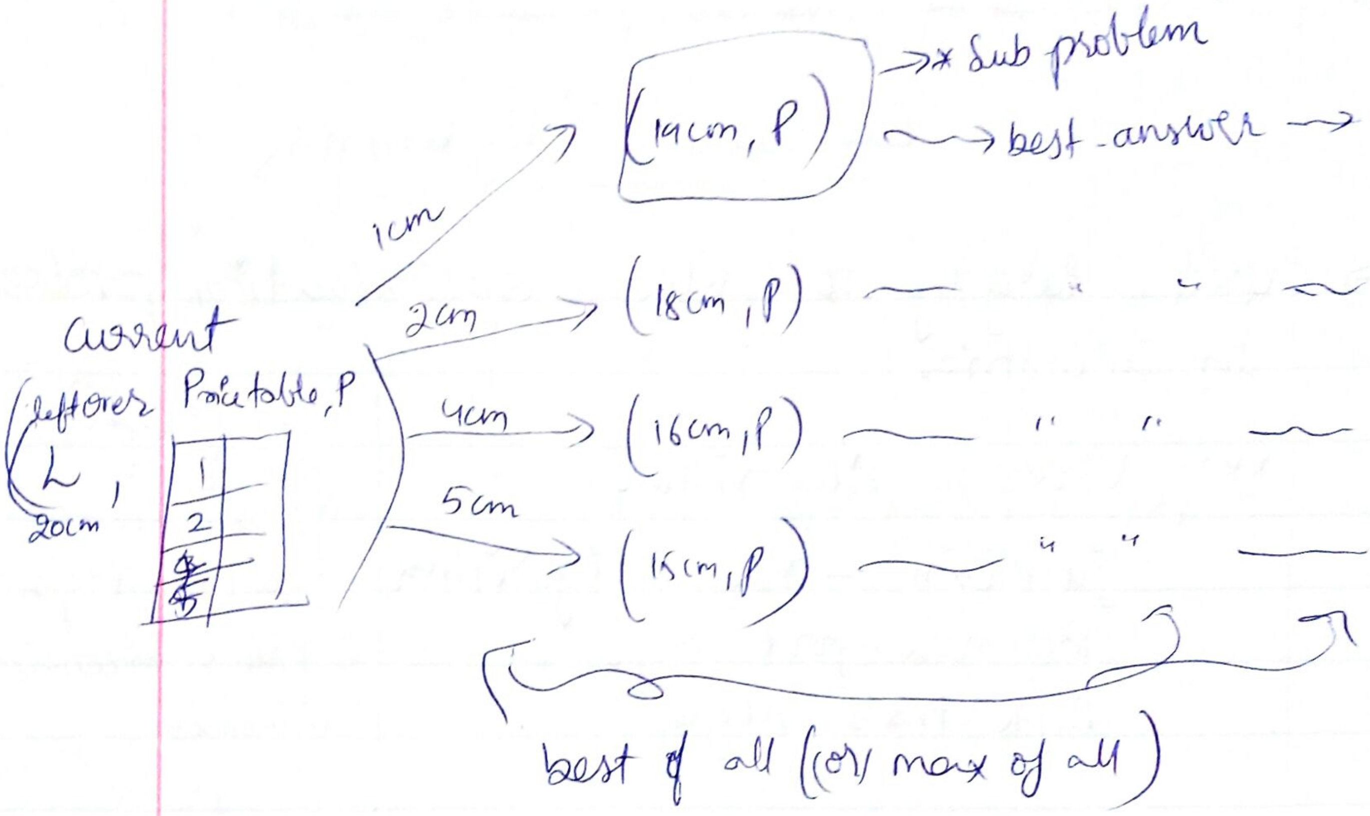
Rod-Cutting Problem:-

cm	$/unit
1cm	20$/unit
2cm	28
4	36
5	45



Optimal Sub-structure:-

① First stage your decision (the first step) and consider left-over part next



Revenue :-

for 1 cm Path : 20¢ + BestRevenue (19, P)
for 2 cm path : 28¢ + BestRevenue (18, P)
for 4 cm path : 36¢ + BestRevenue (16, P)
for 5 cm path : 45¢ + BestRevenue (15, P)

) max is the best for 20 cm.

STEPS AFTER FINDING OPTIMAL SUB-STRUCTURE

1) Write a recurrence

2) Memoize the recurrence

3)

Rod-Cutting Prblm:-

1. Write a Recurrence:

base-cases

$$\text{max\_Rev}(L,P) = \max \begin{cases} 20 + \text{max\_Rev}(L-1,P) \\ 28 + \text{max\_Rev}(L-2,P) \\ 36 + \text{max\_Rev}(L-4,P) \\ 45 + \text{max\_Rev}(L-5,P) \end{cases}$$

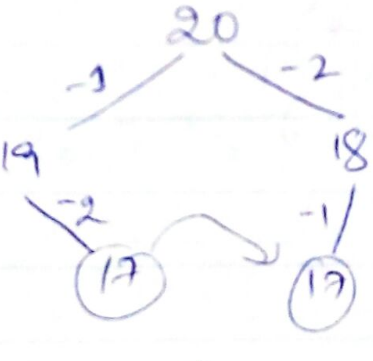
$\text{maxRev}(0,P) = 0$

$\text{maxRev}(L,P) = -\infty$ if $L < 0$

2) Memoizing:

① Top-Down - using the recursion. As you solve a problem, fill it in the hash table.

Ex



→ Look it up, when you are solving a similar problem.

③ Bottom-Up approach:-

→ Solve the problems of smaller size and fill it in the hash table (memoize).

→ Later run the for loop & solve the problem.

$$T[3] = \max \begin{cases} 0 \\ T[0]+1.8 = 1.8 \\ T[-1]+2 = -\infty \end{cases}$$

$$T[4] = \max \begin{cases} 0 \\ T[0]+2 = 2 \\ T[1]+1.8 = 1.8 \end{cases}$$

⋮ ⋮ ⋮ ⋮

→ To recover a solution, we write an another table, which is going to annotate the decision that is giving us the best solution

→ (1) After filling up the decision table, look out for decision at $S[10]$, it gives to cut the rod of length 4.

(2)

S										
NIL	W	W	3	4	4	3				4
0	1	2	3	4	5	6	7	8	9	10

(2) Next, if you cut the rod of length "4", then you'll be left with rod of length '6'.

(3) At '6', the decision is to cut the rod of length "3". If you cut the rod of length "3", then you'll be left with rod of length '3'.

4. Recover the solution:

Ex:

	in	$
l_1	3	1.8
l_2	4	2

$L = 10$

L: Natural numbers

$l_1 \ldots l_n$ } natural numbers

$T[0] = 0$

for $i = 1$ to L

$$T[i] = \max \begin{cases} 0 \\ T[i - l_1] + P_1 \\ T[i - l_n] + P_n \end{cases}$$

$\underline{\underline{L = 10}}$

optimal cost to go

memo-table: T =

<0	0	1	2	3	4	5	6	7	8	9	10
$-\infty$	0	0	0	1.8	2	2	3				4

decision S =

null	waste	waste	3	4	4	3,6				5.6

$$T[1] = \max \begin{cases} 0 \\ T[-2] + 1.8 = -\infty \\ T[-3] + 2 = -\infty \end{cases}$$

~~RC(1,P)~~

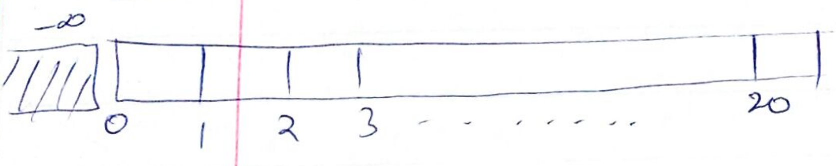
① Rod-cutting(1, P)

② Rod-cutting(2, P)

③ Rod-cutting(4, P)

④ Rod-cutting (5, P)

$\longrightarrow$ memoize table:



for $l = 1$ to L

$$T(L) = \max \begin{cases} 20 + T(L-1) \\ 28 + T(L-2) \\ 36 + T(L-4) \\ 45 + T(L-5) \end{cases}$$

4. At ~~3~~ '3', the decision ~~is~~ is to cut the rod of length '3'.

5. ~~So~~ So, finally the decision is $[4 \to 3 \to 3]$