DAA

ASSIGNMENT- 4.

Q1

let
n= number of items
W= maximum weight capaciity of the knapsack

dynamic programming approach builds a table DP [n+1] [W+1] where each entry represents the max value for a given weight & subset of items

Time complexity:
O(n x W)
∵ we compute each DP cell once

Space complexity:
O(n x W)
if we optimise using 1D array then
O(W)

Q2

APPLICATIONS:

1) Budget Allocation / Investment Planning:
companies can decide how to allocate a fixed budget among multiple projects to maximise profit where each project has a cost (weight) & expected return (value)

Classmate
Date :
Page :

2. Cargo Loading & logistics:
used to determine which items to load into a container or truck with limited capacity to maximise total value without exceeding the weight limit.

Q3

Item	Weight	Benefit
0	2	3
1	2	7
2	4	2
3	5	9

Max Wt (W) = 10
No. of items (n) = 4

Step 1: Initialise DP Table
we create table K[n+1][W+1]
where K[i][w] = max benefit using first i items with capacity w

we fill it using above formula

$$K[i][w] = \begin{cases} 0 & \text{if } i = 0 \text{ or } w = 0 \\ K[i-1][w] & \text{if weight}[i-1] > w \text{, else} \\ \max(K[i-1][w],\ \text{benefit}[i-1] + K[i-1][w - \text{weight}[i-1]]) \end{cases}$$

Step 2: Fill DP Table

- For each item & weight
→ if item too heavy → use previous value
→ else take max of including or excluding the item

- last cell = max benefit.

DP Table

Item/ Capacity	0	1	2	3	4	5	6	7	8	9	10
0	0	0	3	3	3	3	3	3	3	3	3
1	0	0	7	7	10	10	10	10	10	10	10
2	0	0	7	7	10	10	10	10	10	10	10
3	0	0	7	7	10	10	10	10	10	10	10

Max benefit = 19

Selected items: 0, 1, 2 (Total Weight = 9)