

UNIT III DYNAMIC PROGRAMMING:

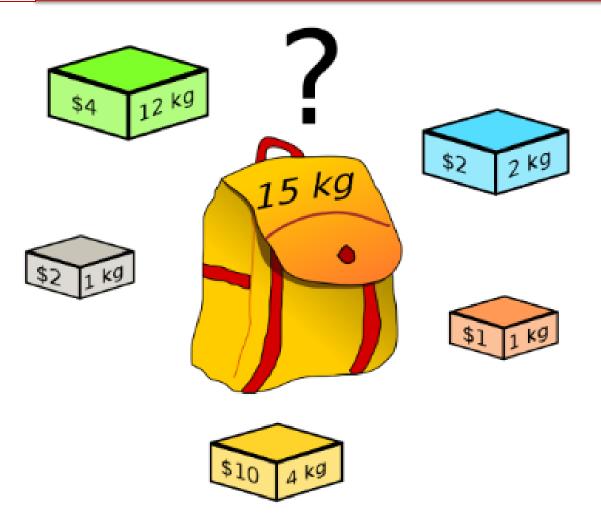
- Dynamic Programming Change-making Problem –
- Computing a Binomial Coefficient All-pairs Shortest-
- paths Problem Warshall's and Floyd's Algorithms –
- 0/1 Knapsack Problem



School of Computing Science and Engineering

Course Code : BSCS2315

Course Name: DAA



Knapsack Problem

Program Name: B.Sc., Computer Science



The Knapsack Problem The classic Knapsack problem is:

A thief breaks into a store and wants to fill his knapsack of capacity K with goods of as much value as possible.

Decision version: Does there exist a collection of items that fits into his knapsack and whose total value is >= W?

0-1 Knapsack: An item can either be picked or left. It cannot be picked partially. For example gold coins, diamond rings, TV etc.



0-1 Knapsack Problem

value[] = {60, 100, 120}; weight[] = {10, 20, 30}; W = 50;

Solution: 220

Weight = 10; Value = 60; Weight = 20; Value = 100; Weight = 30; Value = 120; Weight = (20+10); Value = (100+60); Weight = (30+10); Value = (120+60); Weight = (30+20); Value = (120+100); Weight = (30+20+10) > 50



Knapsack Problem by DP

Given *n* items of

integer weights: $w_1 \ w_2 \ \dots \ w_n$ values: $v_1 \ v_2 \ \dots \ v_n$

a knapsack of integer capacity \boldsymbol{W}

find most valuable subset of the items that fit into the knapsack

Consider instance defined by first *i* items and capacity j ($j \le W$). Let V[i,j] be optimal value of such instance. Then $V[i,j] = \begin{cases} \max \{V[i-1,j], v_i + V[i-1,j-w_i]\} & \text{if } j - w_i \ge 0 \\ V[i-1,j] & \text{if } j - w_i < 0 \end{cases}$

Initial conditions: V[0,j] = 0 and V[i,0] = 0



Course Code : BSCS2315

Course Name: DAA

Knapsack Problem by DP (pseudocode)

```
Algorithm DPKnapsack(w[1..n], v[1..n], W)
   var V[0..n,0..W], P[1..n,1..W]: int
   for j := 0 to W do
         V[0,j] := 0
   for i := 0 to n do
                                       Running time and space: O(nW).
       V[i,0] := 0
   for i := 1 to n do
        for j := 1 to W do
                  if w[i] \leq j and v[i] + V[i-1,j-w[i]] > V[i-1,j] then
                           V[i,j] := v[i] + V[i-1,j-w[i]]; P[i,j] := j-w[i]
                  else
                           V[i,j] := V[i-1,j]; P[i,j] := j
   return V[n,W] and the optimal subset by backtracing
```



Exam	ple: Knap	sack	of c	apa	city	<i>W</i> =	= 5			
item	weight	<u>valı</u>	<u>1e</u>							
1	2	\$1	2							
2	1	\$1	0							
3	3	\$2	0							
4	2	\$1	5			capa	ncity	j		
				0	1	2	3	4	5	
			0	0	0	0				
	$w_1 = 2, v_1$	=12	1	0	0	12				
	$w_2 = 1, v_2$		2	0	10	12	22	22	22	В
	$w_3 = 3, v_3$		3	0	10	12	22	30	32	-
	$w_4 = 2, v_4$		4	0	10	15	25	30	`37	oj i.e

Backtracking finds the actual optimal subset, i.e. solution



Course Code : BSCS2315

Course Name: DAA

Item#	Weight (Kg)	Value (Rs.)
1	2	3
2	3	4
3	4	5
4	5	6

Program Name: B.Sc., Computer Science

Course Name: DAA

Step	Calculation			1	Fable			
1	Initial conditions:							
	$V[0, j] = 0$ for $j \ge 0$	V[i,j]	j=0	1	2	3	4	5
	$V[i, 0] = 0$ for $i \ge 0$	i=0	0	0	0	0	0	0
		1	0					
		2	0					
		3	0					\square
		4	0					
2	W1 = 2,		·	·				
-	Available knapsack capacity = 1	V[ij]	j=0	1	2	3	4	5
	W1>WA, CASE 1 holds:	i=0	0	0	0	0	0	0
	V[i, j] = V[i-1, j]	1	0	0				
	V[1,1] = V[0,1] = 0	2	0					
		3	0					
		4	0					
3	W1 = 2,	-						
3	Available knapsack capacity = 2	V[ij]	j=0	1	2	3	4	5
	W1 = WA, CASE 2 holds:	i=0	0	0	0	0	0	0
	$V[i, j] = max \{ V[i-1, j],$	1	ŏ	0	3		~	–
	vi +V[i-1, j - wi]}	2	0	r -		<u> </u>	<u> </u>	+
	$V[1,2] = \max \{ V[0,2],$	3	0		-		-	\vdash
	3 +V[0,0]}	4	ő	-	<u> </u>	<u> </u>		+
	$= \max \{0, 3+0\} = 3$	<u> </u>						
4	W1 = 2,					-		
	Available knapsack capacity =	V[ij]	j=0	1	2	3	4	5
	3,4,5		0	0	0	0	0	0
	W1 < WA, CASE 2 holds: V[i, j] = max { V[i-1, j],	1	0	0	3	3	3	3
	$v_{[1,j]} = \max\{v_{[1-1,j]}, v_{1} + V[i-1,j], v_{1} + V[i-1,j-w_{1}]\}$	2	0					\square
	$V[1,3] = max \{ V[0,3],$	3	0					
	3 + V[0, 1]	4	0					
	$= \max \{0, 3 + 0\} = 3$							
5	W2 = 3,							
	Available knapsack capacity = 1	V[ij]	j=0	1	2	3	4	5
	W2>WA, CASE 1 holds:	i=0	0	0	0	0	0	0
	V[i, j] = V[i-1, j]	1	0	0	3	3	3	3
	V[2,1] = V[1,1] = 0	2	0	0				
		3	0					
		4	0					
							I	

Program Name: B.Sc., Computer Science

Course Name: DAA

6	$W_2 = 3$,	
	Available knapsack capacity = 2	V[i,j] j=0 1 2 3 4 5
	W2>WA, CASE 1 holds:	i=0 0 0 0 0 0 0
	V[i, j] = V[i-1, j]	1 0 0 3 3 3 3
	V[2,2] = V[1,2] = 3	
		3 0
		4 0
7	W2 = 3,	
	Available knapsack capacity = 3	V[i,j] j=0 1 2 3 4 5
	W2 = WA, CASE 2 holds:	i=0 0 0 0 0 0
	$V[i, j] = max \{ V[i-1, j],$	1 0 0 3 3 3 3
1	vi +V[i-1, j - wi] }	2 0 0 3 4
1	$V[2,3] = \max \{ V[1,3], 4 + V[1,0] \}$	3 0
1	4 + v[1, 0] = max { 3, 4 + 0 } = 4	4 0
8	$= \max\{3, 4+0\} = 4$ W2 = 3.	
°	$w_2 = 5$, Available knapsack capacity = 4	V[i,j] j=0 1 2 3 4 5
	W2 < WA, CASE 2 holds:	i=0 0 0 0 0 0 0
	$V[i, j] = max \{V[i-1, j],$	
	vi +V[i-1, j - wi]}	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
	$V[2,4] = max \{ V[1,4],$	3 0 3 4 4
	4 + V(1, 11)	4 0
	$= \max \{3, 4+0\} = 4$	
9	W2 = 3,	
	Available knapsack capacity = 5	V[i,j] j=0 1 2 3 4 5
	W2 < WA, CASE 2 holds:	i=0 0 0 0 0 0 0
	$V[i, j] = max \{ V[i-1, j],$	1 0 0 3 3 3 3
1	vi +V[i-1, j - wi] }	2 0 0 3 4 4 7
	$V[2,5] = \max \{ V[1,5],$	3 0
1	4+V[1,2]}	4 0
	$= \max \{ 3, 4+3 \} = 7$	
10	W3 = 4	
1.0	Available knapsack capacity =	V[i,j] j=0 1 2 3 4 5
1	Available knapsack capacity = 1,2,3	i=0 0 0 0 0 0 0
1	W3>WA, CASE 1 holds:	
1	V[i, i] = V[i-1, i]	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
1		3 0 0 3 4 4 7
1		
1		
N		

Program Name: B.Sc., Computer Science

Course Name: DAA

11 $W3 = 4$, Available knapsack capacity = 4 W3 = WA, CASE 2 holds: $V[i,j] = max \{V[i+1,j],$ $vi + V[i+1,j - wi]\}$ $V[3,4] = max \{V[2,4],$ $5 + V[2,0]\}$ $= max \{4, 5 + 0\} = 5$ $V[i,j] = 0$ 1 2 3 4 5 12 $W3 = 4$, Available knapsack capacity = 5 W3 < WA, CASE 2 holds: $V[i,j] = max \{V[i+1,j],$ $vi + V[i-1,j - wi]\}$ $V[3,5] = max \{V[2,5],$ $5 + V[2,2]\}$ $= max \{7, 5 + 0\} = 7$ $V[i,j] = 0$ 1 2 3 4 5 13 $W4 = 5$, Available knapsack capacity = V[i,j] = V[i-1,j] $V[i,j] = 0$ 1 2 3 4 5 13 $W4 = 5$, Available knapsack capacity = V[i,j] = V[i-1,j] $V[i,j] = 0$ 1 2 3 4 5 14 $W4 = 5$, $W4 = 5$, $V[i,j] = V[i-1,j]$ $V[i,j] = 0$ 1 2 3 4 5	**	W3 = 4,							
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			VIII	i-0	1	2	3	4	5
$V[i, j] = \max \{V[i-1, j], \\vi + V[i-1, j - wi]\} \\V[3,4] = \max \{V[2, 4], \\5 + V[2, 0]\} \\= \max \{4, 5 + 0\} = 5$ $12 W3 = 4, \\Available knapsack capacity = 5 \\W3 < WA, CASE 2 bolds: \\V[i, j] = \max \{V[i-1, j], \\vi + V[i-1, j - wi]\} \\V[3,5] = \max \{V[2, 5], \\5 + V[2, 1]\} \\= \max \{7, 5 + 0\} = 7$ $13 W4 = 5, \\Available knapsack capacity = 1,2,3,4 \\W4 < WA, CASE 1 bolds: \\V[i, j] = V[i-1, j]$ $\frac{V[i, j] \neq 0}{1} 1 2 3 4 5 7 1 \\4 0 1 1 1 1 \\2 0 0 3 3 3 3 3 1 \\2 0 0 3 4 4 7 1 \\3 0 0 3 4 4 7 1 \\3 0 0 3 4 5 7 1 \\4 0 1 1 0 0 3 3 3 3 3 1 \\4 0 1 1 0 0 3 3 3 3 3 1 \\4 0 1 1 0 0 3 3 3 3 3 1 \\4 0 1 1 0 0 3 3 3 3 3 3 1 \\4 0 1 1 0 0 3 3 3 3 3 3 1 \\4 0 1 1 0 0 3 3 3 3 3 3 3 3$					0	0		0	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$				~	~	-			~
$V[3,4] = \max \{V[2,4], \\ 5+V[2,0]\} = \max \{4,5+0\} = 5$ $V[3,4] = \max \{4,5+0\} = 5$ $V[3,4] = \max \{4,5+0\} = 5$ $V[2,0]\} = \max \{4,5+0\} = 5$ $V[1,j] = \max \{1,j], \\ V[3,5] = \max \{V[2,5], \\ 5+V[2,1]\} = \max \{7,5+0\} = 7$ $V[1,j] = V[1,j]$		vi +V[i-1, j - wi]}		-	~	-	-	-	
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$				-					<u> </u>
$ = \max \{4, 5+0\} = 5 $ $ 12 W3 = 4, $ Available knapsack capacity = 5 $ W3 < WA, CASE 2 \text{ holds:} \\ V[i, j] = \max \{V[i-1, j], \\ vi + V[i-1, j - wi]\} \\ V[3,5] = \max \{V[2, 5], \\ 5 + V[2, 1]\} \\ = \max \{7, 5+0\} = 7 $ $ 13 W4 = 5, $ Available knapsack capacity = $ 1,2,3,4 \\ W4 < WA, CASE 1 \text{ holds:} \\ V[i, j] = V[i-1, j] $ $ V[i, j] = V[i-1, j] $				~	· ·	5	-	3	\vdash
Available knapsack capacity = 5 $V[i,j]$ $j=0$ 1 2 3 4 5 $V[i,j]$ $= max$ { $V[i-1, j]$, $vi + V[i-1, j-wi]$ } $vi + V[i-1, j-wi]$ } 1 0		$= \max \{4, 5 + 0\} = 5$		U					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	12	,							
$V[i, j] = \max \{V[i-1, j], \\vi + V[i-1, j - wi]\} \\V[3,5] = \max \{V[2, 5], \\5 + V[2, 1]\} \\= \max \{7, 5 + 0\} = 7$ $13 W4 = 5, \\Available knapsack capacity = \\1,2,3,4 \\W4 < WA, CASE 1 holds: \\V[i, j] = V[i-1, j]$ $\frac{V[i, j] = V[i-1, j]}{3 0 0 3 4 5 7} \\V[$			V[i,j]	j=0	1	2	3	4	5
$V[3,5] = \max \{V[2,5], \\ 5+V[2,1]\} \\ = \max \{7,5+0\} = 7$ $V[i,j] = V[i-1,j]$ $V[3,5] = V[i-1,j]$ $V[i,j] = V[i-1,j]$				0	0	~		- C	~
$V[3,5] = \max \{V[2,5], 5+V[2,1]\} = \max \{7,5+0\} = 7$ $13 W4 = 5, Available knapsack capacity = 1,2,3,4 W4 < WA, CASE 1 holds: V[i,j] = V[i-1,j]$ $V[i,j] = V[i-1,j]$			-	~	0		-		_
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$					~	~	-		_
$= \max \{7, 5 + 0\} = 7$ $13 W4 = 5,$ Available knapsack capacity = 1,2,3,4 W4 < WA, CASE 1 holds: V[i, j] = V[i-1, j]			3	-	0	3	4	5	7
13 $W4 = 5$, Available knapsack capacity = 1,2,3,4 W4 < WA, CASE 1 holds: V[i, j] = V[i-1, j] $V[i,j] = 0$ 1 2 3 4 5 1 0 0 0 0 0 0 0 0 1 0 0 3 3 3 3 3 3 2 0 0 3 4 4 7 3 0 0 3 4 5 7 4 0 0 3 4 5 7			4	0					
Available knapsack capacity = $1,2,3,4$ W4 < WA, CASE 1 holds:		$= \max\{7, 5 \neq 0\} = 7$							
i=0 0 0 0 0 0 0 $W4 < WA$, CASE 1 holds: $i=0$ 0 0 0 0 0 0 $V[i, j] = V[i-1, j]$ $i=0$ 0 0 3 3 3 3 2 0 0 3 4 4 7 3 0 0 3 4 5 7 4 0 0 3 4 5 7	13	W4 = 5,							
W4 < WA, CASE 1 holds: $V[i, j] = V[i-1, j]$ 1 0 0 3 3 3 2 0 0 3 4 4 7 3 0 0 3 4 5 7 4 0 0 3 4 5 7		Available knapsack capacity =	V[i,j]	j=0	1	2	3	4	5
V[i, j] = V[i-1, j] $2 0 0 3 4 4 7$ $3 0 0 3 4 5 7$ $4 0 0 3 4 5$			i=0	0	0	0	0	0	0
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$			1	0	0	3	3	3	
4 0 0 3 4 5		V[i, j] = V[i-1, j]	2	0	0	3	4	4	7
			3	0	0	3	4	5	7
14 W4 = 5,			4	0	0	3	4	5	
	14	W4 = 5,							
Available knapsack capacity = 5 $V[i,j]$ j=0 1 2 3 4 5		Available knapsack capacity = 5	V[i,j]	j=0	1	2	3	4	5
W4 = WA, CASE 2 holds: i=0 0 0 0 0 0 0 0			i=0	0	0	0	0	0	0
$V[i, j] = max \{ V[i-1, j], 1 0 0 3 3 3 3$	1 1	$V[i, j] = max \{V[i-1, j],$	1	0	0	3	3	3	
		vi +V[i-1, j - wi] }	2	0	0	3	4	4	
		vi + V[i-1, j - wi] V[4,5] = max { V[3, 5],		~	-		-		
$= \max\{1, 0+0\} = 1$		$vi + V[i-1, j - wi] \}$ V[4,5] = max { V[3, 5], 6 + V[3, 0] }	3	0	0	3	4	5	7
Maximal value is V [4, 5] = 7/-		vi + V[i-1, j - wi] V[4,5] = max { V[3, 5],	3	0	0	3	4	5	7

Program Name: B.Sc., Computer Science

Course Name: DAA

Step	Table	Remarks
1	V[i,j] j=0 1 2 3 4 5 i=0 0 0 0 0 0 0 0 1 0 0 3 3 3 3 3 2 0 0 3 4 4 7 3 0 0 3 4 5 7 4 0 0 3 4 5 7	V[4,5]=V[3,5] → ITEM 4 NOT included in the subset
2	V[i,j] j=0 1 2 3 4 5 i=0 0 0 0 0 0 0 0 1 0 0 3 3 3 3 3 2 0 0 3 4 5 7 3 0 0 3 4 5 7 4 0 0 3 4 5 7	V[3,5]=V[2,5] → ITEM 3 NOT included in the subset
3	V[i,j] j=0 1 2 3 4 5 i=0 0	V[2,5] ≠ V[1,5] → ITEM 2 included in the subset
4	V[i,j] j=0 1 2 3 4 5 i=0 0 0 0 0 0 0 0 1 3	V[1,2] ≠ V[0,2] → ITEM 1 included in the subset
5	Since item 1 is included in the knapsack: Weight of item 1 is 2kg, therefore, remaining capacity of the knapsack is (2 - 2 =) 0 kg.	Optimal subset: { item 1, item 2 } Total weight is: 5kg (2kg + 3kg) Total profit is: 7/- (3/- + 4/-)

Program Name: B.Sc., Computer Science

