Experiment. No.04

ritle: write a program to solve a 0-1 knapsack problem using dynamic programming or branch and bound strategy.

objective: To understand and solve 0-1 knapsack problem using dynamic programming

theory:

what is Dynamic Programming:

Dynamic Programming is also used in optimization problems. Like divide and conquer method, Dynamic programming algorithm solves each sub-problems just once by combining the solutions of sub-problems.

oynamic Programming algorithm solves each subproblem just once and then uses its answers in
a stable, thereby avoiding the work of re-computing
the answer everytime.

Two main properties of a problem suggest that the given problem can be solved using Dynamic programming. These properties are overlapping sub-problems & optimal substructure.

For example, Binary search does not have overlapping sub-problem, whereas recursive problems of fibonacci numbers have many overlapping sub-problems.

144.2	steps of Dynamic Programming Approad
	Approad
	It is designed using the following
	3 30/19
2 d 15	1. characterize the structure of an optima 2. Recursively define the value of
	1. Characterize the structure of an optima 2. Recursively define the value of an optima solution.
	solution.
	3. compute the value of the optimal solution
	p p cully 1/1 a half-and
	an notimal a living
£	computed information.
1. 1	Application of
	Application of Dynamic Programming
	riarrix chain multipl'
	- Longest common subsequence - Travelling salesman
	- Travelling salesman Problem
	- ODIEM
	Lha
	knapsack problem!
	knapsack problem! - A knapsack with limit
	knapsack problem! - A knapsack with limited weight capace Yalua
	knapsack problem! - A knapsack with limited weight capace yalue
	Knapsack problem! - A knapsack with limited weight capace value - Few items each having some weight was weight to the capace was weight to the capace weig
	Knapsack problem: - A knapsack with limited weight capace value - Few items each having some weight was weight to the capace was a some weight was a some weight to the capace was a some weight was a some weight to the capace was a some weight was a some weight to the capace was a some weight was a some weight to the capace weight to the capace was a some weight to the capace was a some weight to the capace was a some weight to the capace with the capace was a some weight to the capace was a some was a som
	knapsack problem: - A knapsack with limited weight capace value some weight waving some weight wariants. knapsack problem has the following to the source of the source
	knapsack problem: - A knapsack with limited weight capace value some weight waving some weight wariants. knapsack problem has the following to the fractional knapsack to the following to the first to the
	knapsack problem: - A knapsack with limited weight capace value some weight waving some weight wariants. knapsack problem has the following to the source of the source

										_=
0/1	Knapsad	c 15	pro	blem	us,	ing	Gri	eedy	method	
			4		<u> </u>	<u></u>		, L		
AND DESCRIPTION OF THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS NAMED IN COLUMN TWO I	sider	3 1		Lings of Lond		1 1 1 3	1			
15	napsack	weig	phr	Capa	city	= U	ر	1	116 2 110-1	
N	o of ite	ems	ea	.ch	havi	ng	SOM	e cé	seight & value=n	
										—Ţ-
step	1 . Dr	aw c	a > t	able	say	1	73	WIFF	h (ntil no of.	
	_	N /		l	κ,	~ ^	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	h /		
	Fill all	the	<u> </u>	oxes	OF) +0	rou	of oth column	
ωi	th zeros		Aut	4 10 3	\$ 1 th +		Cirr		realC	
	John Branch	133	T. V.	2.3			m. A	ω	V 152, 25 ft	
	4.7. 1.00	0			3		5 () (PTRAIN	
	0	0	0	0	G	pegr	in 7/ -	-	0.5 (2)	
	1	0	-							
	2	6	-		The state of the s		11.803	19000	3(11)	
						-				
* ,	b	0	134	Ste	\ *	Q :		0 0 74 1		
	<u> </u>		(to)	= 5	11)		:	
	21.16	V 7.5	- (T-ta	ble	4 1	1 1/2 =	373110	131	
	<u> </u>		- Am - A	. ()	1.010		11-	2.0	12 12100 +00 +0	
step	2 - sta	ert	<u> Filli</u>	ng	+ne	<u>ta</u>	<u>610</u>		w wise top to	
	bottom	from	le	Ft F	0 7	ight		110	I-ulana)	
		1775		_						
	use the	Follo	Wing	9 FO	smul	á –			11 11 11 11 11 11 11 11 11 11 11 11 11	
		, , ,				<i>(</i>		1	1.10 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	_
	TCI	, U)	= M	ear	{ T	<u> </u>	<i>i</i> J	V	alue ; + T (i-1, J	
									weight; 13	
				THE STREET STREET, STR						
	This s	tep l	eade	s to	COM	ple	rlty	f	illing the table	2
	Then, 1	value	0	F +>	ne 1	ast	bo	α	represents the	
maz	cimum po	ossib	le	valu	le	that	CC	an b	e put into the	3

knapsack. step 3 - To identify the items that put into the knapsack to obtain the profit. - consider the last column of the table - start scanning the entries from bottom to - on encountring an entry where value is as the value stored in the entry immedia above it, make the row label of that entry - After all the entries are scanned, then labels represent the items that must be into the knapsack. Time complexity: It takes o(n w) time to fill (n+1) (n+1) to

It takes o(n) time for taking the solut sing process traces the rows

Thus overall o (nw) time is taken to so taking process traces the In this way, we have studied of the conclusion -