H.W-6

Name: JRI MANVITH VADDEBOYINA USC: 1231409457

algorithm to compute the optimal value you can get from a knapsack with capacity W.

Given infinitely number of term of each to type, the problem is similar to knowsack problem and aptimal solution is given by,

OPT (1, w) - max S OPT (2-1, w),
and OPT (0,0) = 0 OPT (1, w-w);) + V;
As there are infinitely many number of tems
of each type, we choose another tem of type;
and proceed with subproblem.
We have two options whether to choose to
the item (or) not. Based on this we have
obtained the above recurrence relation.

This would give us the optimal solution & you can get from a knapsack of Capacity W.

Q. OPT $(K) = \int Max(OPT(i))$, $0 \le i \le k$ and $s_{i+1}^* k$ is a word in diensonary opt(0)=0 0, otherwise.

Let Si,k denote substring Si Si+1-Sk.

OPT(K) =1 -> segmentation is possible

O => Otherwise.

Segmentation is possible if only the last word is it is the dictionary and the Gremaning substrug can be segmented.

Computing the value of OPT (O), --- OPT (n) using the above relation would give us the OPT (n) which is our solution.

This can be computed en O(12) time.

'n baloons indexed 0-7n-1 nums [-1]= nums [n]=1 Design a dynamic programming algorithm to find the maximum coins you can collect by buriting the balloons wisely. Avalyze over time Implesely Colilias. For this problem we have recurence relation as, of this production.

OPT $(l,r) = max \begin{cases} oPT (l,k-1) + OPT (k+1,r) \\ nums [k] * nums [l-1] \\ * nums [r+1] \end{cases}$ dp[N+1][N+1] BURST_BALLOOMS (J, 92): IF LYR: MAX-COST = -INT-MAX RETURN O JE (de[I][]] !=-1) FOR EACH K:L -> R: COST = nums[d-1] * nums[x] * nums[x+1] BURST-BALLOONS (l, K-1) BURST _ BALLOONS (K+1, 2) MAX_COST = MAX (MAX_COST, COST) ENDFOR THE COST dp[l][x] = MAX_cosT veturn dp[1][n] Time Complesety & T.C = N X N X N = N3 1 1 1 2 states loop T. C = 0 (N3)

Devise a Dynamic Programming algorithm to determine the maximum amount of money you can get by cutting the rod strategically and selling the Given a vod can be cut into pieces. Let the length of pieces be 0 -- N'i.e length [0,+;--N] Cost of each wood of length ? is pi dollars.
Cost of each piece of length i. Let itte optimal solution be OPT OPT (i, k) = { costt i-1] +OPT (i, k-length [i-1] of cesolution copT (i-1,k) This problem is similar to unbounded knapsack problem with weights = lengths values = costs W = N Using memoization, dp[N+1] [N+1], where dp[0][0]=0, others=-1. AL GORITHM: ROD_BP (length, cost, n, N): IF n=0 or N=0: return 0

IF lengthtm-IJ <= N:

dp[n][N]= masc (cost[n-I] + ROD-DP (

dp[n][N]= masc (cost[n-I] + ROD-DP (

length[n-I])

ALGORITHM	
ROD-DP (deugth, cost,	n,N):
1F n=00R 1	J=O: RETORNO
dptn/[N] =	$J \leftarrow N_3$ made (cost [n-1] +
	ROD_DP (length, cost, n,
	N-Cenget [n-1]
	ROD-DP (length, cost, n, N- Congeth [n-1]), ROD-DP (length, cost, n-1, N);
ELSE:	
dptnj[N] = ROD_DP (dength, coit, n-1, N)
return de [m][N] = ROD - DP (dength, cost, n-1, N)
regional contraction of the second contracti	
algorium give	s us the maximum amount.

5. For this problem we have the vecurence relation as $OPT(n) = min S_{\ell,n}^2 + OPT(j-1)$ $1 \le j \le n$

The clast cline of words wi, -- wn is used in an aptimum solution if and only if minimum is obtained using index j.

ALGORITHM:

Sij= 10 if words exceed total length L.

0=[0] T90

FOR k=1,--n

OPT[1] = min (Sj,k + OPT (j-1))

1=j=k

END FOR Return OPTENI

Time Complexity of

Each "teration: O(n), n éterations > O(n) * n. T. C= O(n2)

6.	a) The 1	selow exam	fle will result	= %	in correct	answer	
			Minute 2				2
							S
	В	1	10		1		
	The gu	en algorit	tun will fol	llow (the sequ	ience	
	of 2 +	1,00 / a3	tun will fol ie AAA 10	hich	gives r	ralue of	
	But	optimal	solution was	eld be	L BBB,	shich	
	would	gue	solution was	10 +1 =	12.		
							6
and the second s	P) Subso	se we have	optimal plans) and ending	for	sequences	upto	6
	min	ute.) and ending	wu	n A w V	ne Clast	•
	Let	it be de	noted by Mox of optimal plans and Max CB	CAJC	?-2] Sin	ularly _	6
gar tear i kasari ng dhig aga kasarin yan	Max.	CB7[1-1]	and Max [B	ns end	ling & B	twely	•
		The second secon				and the second s	•
Dengago en Lo etc comprehense		on other	we can find	m	AJJCA Jan	DINO WAS	Ade Co
transaction to all territorists	y Andrew (no. 1970) and to recent control (1987) y the filteration (1987) and	and the second s	6	1			-
	Maxt	= [i][A]	max (Maxte			N	
		and the second second second second	N	Nask [B][1-2]		
namentos el gertanes a nino.	Mark	[1][8]	= mase (Ma	UX[B	1[1-1]+	bi,	
antiga e un representante de la composición del composición de la composición de la composición del composición de la composición del composición del composición del composición del composició					Marset	A][1-2]	
epuntaria de la como d	The	0 0 0	0	. / ~	A TITATION	7	
ance are all the rear 18 Miles	J.W (primar n	ralue = mas	x (11	Marx	CBJ[:7)	
re place - A filmana, majorità e	an and Market Andrews (Allegeria) and the Andrewski god, any state of the second control	and the second of the second o	THE NEW TO ANGLE SERVICES IN THE PROPERTY OF T		g g g g g g g g g g g g g g g g g g g		
							400

ALGORITHM: Max [A][O];=0 and Max[B][O];=0 Max [A][i]:=ai, Max[B][i]:=bi Set 1:=2 WHILE IC= n IF MAX CAJEI-17+ai < MONEBJEI-2] Max [B-i] [B] snaM =: [i][A] xnaM ELSE Max[A][i]:= Max [A][i-1] +ai ENDIF Replace aiby b? incoment? END WHILE RETURN MON(MAX [A][n], MAX[B][n]) END $T_{\circ}C = O(n)$