DAA LAB-05:

(A) PRACTIONAL KNAPSACK. Apprithm.

(1) Algorithm

11 This function calculates plus rotto for each item 11 Input: Dataset array of form [val, wt, sney wife] 11 Output: None.

act prow_ratio (items):

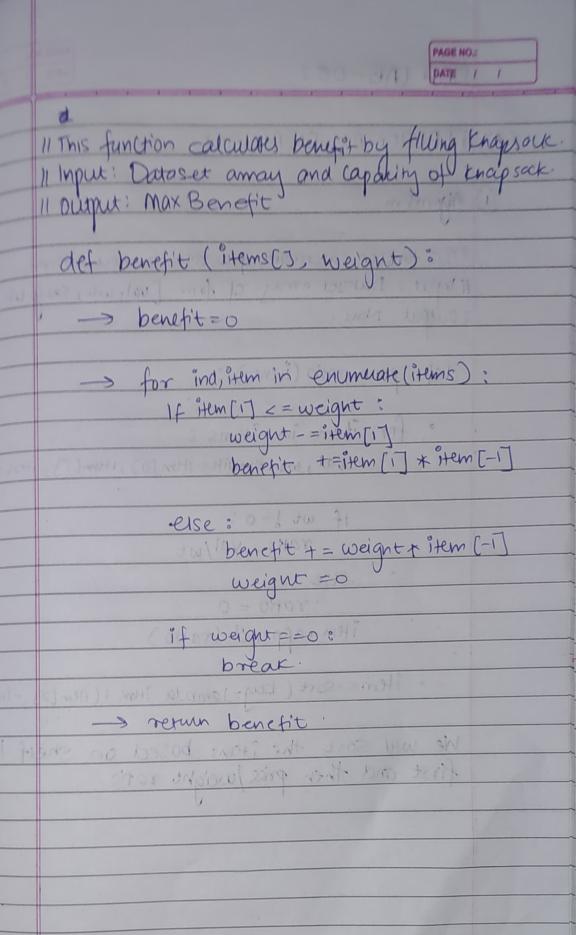
· for item in items: val, wt, shellife= item [0], item [1], item [2]

> If wt 1=0: 1213. ratio = val/wt

ratio = 0 item append (ratio)

· items. sort (key= lambda item: (item[2], -item[-])

We will sort the items based on shelf life first and then price/weight rate.



DATE /	PAGE	NO.:
	DATE	1

(2) Testcases

Positive TCS:
Different Testcases with valid Value of all fields in input.

Display maximum benefit along with the items added in the knapsock

Negative TCS:

1. Negative Price.

For eq. items [3] = [-100, 25, 10]

Display -> Price Can't be negative

(2.) Weight negative

For eq. items [7] = [1000, -3, 4]

Display => weight (an't be negative

3.) Negarire snelflife.

For eq. items [6] = [400, 8, -19]

Display => Shelf life can't be negative.

Dotaset Empty

For eg. items = []

Display - Dataset is empty

		PAGE NO.:
	8740	DATE / /
(3)	Time Complexity.	
(3)	27/93	1-1
	BRUTE FORCE:	
	To solve by brute force we need	to consider
	au the subsets of items we	
	the number of subsett of a set	
	How ever sonce this is fractional	anapsade,
-	we can take prochions lof items.	0.1
	: Time complexity = 0 (K") wh	rece,
	K is the number of frac	thous of an item
		1
	Leganse Proc.	
	GREEDYZ: 001-7 = [8] 20171 10 201	have
	By solving by greedy approach	we have
	3 functions in the program:	
	Martin the or	may once
	check():- Traverses through the or and checks if all values or	re valid.
	and checks it act of (n)	
	Time Compunity - O(n)	
3	liver through the	omay
	prow-ratio():- Herates through the and calculates plis ratio	> 0 cms
	The sould be seed	4.240 1.00
•	Sorts the anay based	on street ut
	sorts the away based and pluration -> c	(nwgn)
	2413	Concess only
	Benefit () - Herates through the	array or ay
	ona soin)	1,000
	: TC = n + nlogn +	+n wyn
	tc = O(nwgn)	

Л