CSE 2320 - Homework 6 (You can solve all of P3 and parts of Knapsack. Monday Oct 22, we will cover Greedy and 0/1 for Knapsack.)

NAME: GOUTAMI PADMANABHAN UTA ID : 1001669338

Total points: 115/100 (points past 100 are bonus) Topics: Memoization, Greedy, Dynamic Programming (Knapsack: unbounded, 0/1, fractional)

P1 (4 pts) Given this solution information, for the **unbounded** Knapsack problem below, recover the choices that gave the optimal answer for knapsack capacity 19. Show your work (highlight or circle cells).

Item	A	B	C	D
Weight	3	4	7	8

Value | the item values are hidden as they should not be used in recovering the solution.

picked	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
				A	B	B	A	C	D	D	A	B	B	A	C	C	D	A	B	B

Items picked for capacity 19: ....D, C, B.....

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
picked				A	B	B	A	C	D	D	A	B	B	A	C	C	D	A	B	B

Maximum capacity = 19

For size 19, B is the picked item

Remaining weight = 19 - Weight of B

= 19 - 4 = 15

For size 15, C is the picked item

Remaining Weight = 15 - Weight of C

= 15 - 7 = 8

For size 8, D is the picked item

Remaining weight = 8 - Weight of D

= 8 - 8 = 0

For size 0, there is no picked item

Items picked for capacity 19 = D, C, B

P2 (61 pts) Given the item types below, solve the following problems. Fill in the answer in the table and show your work below.

Item:	A	B	C	D
Weight:	3	4	6	7
Value:	4	7	10	12

	Unbounded Knapsack	0/1 Knapsack	Fractional Knapsack
Dynamic Programming	\$\$: 24 Items: C, B, B	\$\$: 23 Items: A, B, D	
Greedy	\$\$: 21 Items: B, B, B	\$\$: 23 Items: B, D, A	\$\$: 24 Items: B, D, $\frac{1}{2}$ of C

a) (20 pts) Solve the **unbounded** Knapsack problem. Recover the items in the solution and show how you did that (e.g. highlight or circle cells). Show your work as done in class.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Sol	0	0	0	4	7	7	10	12	12	14	17	19	20	22	24
Picked	-	-	-	A	B	B	C	D	D	A	B	B	C	C	B
A, 3, 4	-	-	-	0, 4	1, 4	2, 4	3, 8	4, 11	5, 11	6, 14	7, 16	8, 16	9, 18	10, 21	11, 23
B, 4, 7	-	-	-	-	0, 7	1, 7	2, 7	3, 11	4, 11	5, 14	6, 17	7, 19	8, 19	9, 21	10, 24
C, 6, 10	-	-	-	-	-	-	0, 10	1, 10	2, 10	3, 14	4, 17	5, 17	6, 20	7, 22	8, 22
D, 7, 12	-	-	-	-	-	-	-	0, 12	1, 12	2, 12	3, 16	4, 19	5, 19	6, 22	7, 24

Maximum capacity = 14

For size 14, B is the picked item

Remaining Weight = 14 - Weight of B

= 14 - 4 = 10

For size 10, B is the picked item

Remaining Weight = 10 - Weight of B

= 10 - 4 = 6

For size 6, C is the picked item

Remaining Weight = 6 - Weight of C

= 6 - 6 = 0

For size 0, there is no picked item

Items picked for capacity 14 : C, B, B

Total value for the items picked : 10 + 7 + 7 = 24

b) (20 pts) Solve the 0/1 knapsack problem below (15pts). Use a star to show if the current item was used or not in the solution (8pts). Recover the items in the solution and show how you did that (e.g. highlight or circle cells) (7 pts). Show your work as done in class.

	0	1	2	(3)	4	5	6	(7)	8	9	10	11	12	13	(14)
No item	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A, 3, 4	0	0	0	(4*)	4*	4*	4*	4*	4*	4*	4*	4*	4*	4*	4*
B, 4, 7	0	0	0	4	7*	7*	7*	(11*)	11*	11*	11*	11*	11*	11*	11*
C, 6, 10	0	0	0	4	7	7	10*	11	11	14*	17*	17*	17*	21*	21*
D, 7, 12	0	0	0	4	7	7	10	12	12*	14	17	19*	19*	22*	(23*)

Maximum capacity = 14 , Maximum value picked = 23 for item D

Remaining weight = 14 - weight of D

= 14 - 7 = 7

For size 7, 11 is the value for C, but it is not picked

For size 7, 11 is the picked value for B

Remaining weight = 7 - Weight of B

= 7 - 4 = 3

For size 3, 4 is the picked value for A

Items picked for capacity 14 : A, B, D

Total value for the items picked: 4+7+12 = 23

c) (8 pts) What items will a Greedy algorithm based on the **ratio**, choose for an **unbounded** knapsack problem of size 14? Show your work.

Ratio of Value / Weight

Ratio of A = $\frac{A}{3}$ = 1.3

Ratio of B = $\frac{7}{A}$ = 1.75

Ratio of C = $\frac{10}{6}$ = 1.6

Ratio of D = $\frac{12}{7}$ = 1.71

By arranging the items in descending order of their ratio, we get

B, D, C, A

This order will be used in problems

P2) c, d, e

Maximum capacity or problem size = 14

Since B has the highest ratio; for size 14, B is picked. B's weight fits with 14.

Remaining Weight = 14 - Weight of B

= 14 - 4 = 10

B's weight fits with size 10. Again, B is picked

Remaining Weight = 10 - Weight of B

= 10 - 4 = 6

B's weight fits with size 6. Again B is picked

Remaining Weight = 6 - Weight of B

= 6 - 4 = 2

2 does not fit anywhere

Items picked for size 14: B, B, B

Total value for the items picked: 7+7+7 = 21

d) (8 pts) What items will a Greedy algorithm based on the ratio, choose for a 0/1 knapsack problem of size 14? Show your work.

Maximum capacity of problem size = 14

Since, B has the highest ratio; for size 14, B is picked. B's weight fits with 14

Remaining weight = 14 − Weight of B

= 14 − 4 = 10

B's weight again fits with 10. But it has already been chosen.

So choose D - the second highest ratio

Remaining weight = 10 − Weight of D

= 10 − 7 = 3

C has the third highest ratio, but its weight does not fit with 3

So choose A - A's weight fits with 3

Remaining Weight = 3 − Weight of A

= 3 − 3 = 0

0 does not fit anywhere

Items picked for size 14 : B, D, A

Total value for the items picked : 7 + 12 + 4 = 23

e) (5 pts) What items will a Greedy algorithm based on the **ratio**, choose for a **fractional** knapsack problem of size 14? Assume you have only one of each item. Show your work.

Maximum capacity of problem size = 14

Since B has the highest ratio; for size 14, B is picked. B's weight fits with 14

Remaining weight = 14 − Weight of B

= 14 − 4 = 10

B's weight again fits with 10. But it has already been chosen.

So choose D. – the second highest ratio.

Remaining weight = 10 − weight of D

= 10 − 7 = 3

C has the third highest ratio, but only a fraction of its weight fits with 3

$\frac{3}{6}$ of C's weight can be fit in 3 $\Rightarrow \frac{1}{2}$ of C

Items picked for size 14: B, D, $\frac{1}{2}$ of C

7

Total value for the items picked: $7 + 12 + \left(\frac{3}{6} \times 10\right) = 7 + 12 + 5 = 24$

P3 (50 pts) Consider this recursive function:

```
1→int foo(int N){
2→   if (N <= 1) return 5;
3→   int res1 = 3*foo(N/2);
4→   int res2 =   foo(N-1);
5→   if (res1 >= res 2)
6→       return res1;
7→   else
8→       return res2;
  }
```

a) (6 points) Write the recurrence formula for the TIME COMPLEXITY of this function, <u>including the base cases</u> for N>=0. You do NOT need to solve the recurrence. Remember not to confuse the time complexity of the function with what the function calculates.

The recursive call is done only in lines 3 and 4.

$$T(N) = T(N/2) + T(N-1) + c$$

b) (8 points) Draw the tree that shows the function calls performed in order to compute foo(5) (the root will be foo(5) and it will have a child for each recursive call.) Also show what each call returns by using an arrow pointing back from the child to the parent.