CSC2400 / Chapter 8 Homework

Due: Monday, April 20, 2020

NAME:_________________Bradley Harper 002____________________________

1. **(6 points)** What does dynamic programming have in common with divide-and-conquer? What is a principal difference between them?

Each problem is split into sub problems and solved individually. The principal difference is that Dynamic stores the sub problems generally in a table so the sub problems don't need to be solved again, it then uses the table to solve the original problem.

2. **(10 points)** Solve the instance **10, 25, 1, 1, 5, 1, 25, 10, 10** of the coin-row problem using the dynamic programming algorithm technique. Show the solution array and also the final output.

Given: n=9

	1	2	3	4	5	6	7	8	9
C	10	25	1	1	5	1	25	10	10

Step 1:

	0	1	2	3	4	5	6	7	8
A	0	10							

Step 2: A[0] = 0 , A[1] = C[1]

Step 3. $A[I] = \text{Max}\begin{cases} C[I] + A[I-2] \\ \text{or} \\ A[I-1] \end{cases}$

i starts at 2 and goes through n=9

	0	1	2	3	4	5	6	7	8	9
A	0	25	25	28	28	33	33	58	58	68

25+0 or 25

3+25 or 25

1+25 or 28

5+28 or 28

1+28 or 33

25+33 or 33

10+33 or 58

10+58 or 58

Step 4 return A[N]

	0	1	2	3	4	5	6	7	8	9
A	0	25	25	28	28	33	33	58	58	68

→return 68

3. **(10 points)** Using the change making DP algorithm, give change for amount **8** using the minimum number of coins of denominations (**1, 2, 3, 4**). You may assume that there is an unlimited quantity of coins for each of the 4 denominations. Determine how many coins and also which coins will be used to make change.

amount = 8 coins = {1, 2, 3, 4}

i = 0 Step 1 F

0	1	2	3	4	5	6	7	8
0								

F[0] = 0

F[i] = min (F[i-1], F[i-2], F[i-3], F[i-4]) + 1

i = 1

F[i] = F[1-1] + 1 = 1

i = 2

F[i] = F[2-2] + 1 = F[0] + 1 = 1

i = 3

F[1] = F[3-3] + 1 = F[0] + 1 = 1

i = 4

F[i] = F[4-4] + 1 = F[0] + 1 = 1

i = 5 = F[5-1] + 1 = F[1] + 1 = 2

i = 6, F[i] = F[6-2] + 1 = F[2] + 1 = 2

i = 7, F[i] = F[7-3] + 1 = F[3] + 1 = 2

i = 8, F[i] = F[8-4] + 1 = F[4] + 1 = 2

0	1	2	3	4	5	6	7	8
0	1	1	1	1	2	2	2	2

Coin 4 produced F[8] = 2

So we need 2 of coin 4 to get 8

4. Knapsack Problem.
   a. **(15 points)** Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem:

      - item 1, weight is 3 lbs, value is $25
      - item 2, weight is 2 lbs, value is $20
      - item 3, weight is 1 lb, value is $15
      - item 4, weight is 4 lbs, value is $40
      - item 5, weight is 5 lbs, value is $50
      - **Capacity of knapsack is 6 lbs total**

b. **(5 points)** What is the maximum value of a feasible subset of the knapsack in part (a)?

weight

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5		0	0	0	0	0	0

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	25	25	25	25
2	0	0	20	25	25	45	45
3	0	15	20	3.5	40	45	60
4	0	15	20	35	40	55	60
5	0	15	20	35	40	35	65

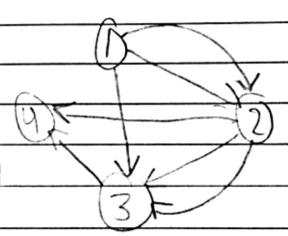
?

$65

5. **(15 points)** Apply Warshall's algorithm to find the transitive closure of the digraph defined by the following adjacency matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$



SCORE: ______ / 61