Assignment on Dynamic programming and Greedy algorithm design paradigms

Problem 1

Find an optimal parenthesization and minimum number of multiplications required of a matrix-chain product for each of the following sequence of dimensions are:
a. < 5, 10, 3, 12, 5, 50, 6>
b. <3,5,4,6>
c. <10,100,5,50>
d. <30,35,15,5,10,20,25>
e. <5,4,6,2>

Problem 2

Run the greedy algorithm for fractional knapsack problem on the following data:

a. n = 4 (number of items)
M = 5 (knapsack capacity = maximum weight)
(wi, pi): (2, 3), (3, 4), (4, 5), (5, 6)

b. n = 4 (number of items)
M = 10 (knapsack capacity = maximum weight)
(wi, pi): (2, 10), (4, 40), (6, 30), (3, 50)

c. n = 3 (number of items)
M = 20 (knapsack capacity = maximum weight)
(wi, pi): (18, 25), (15, 24), (10, 15)

Problem 3

Tradeoff between Dynamic Programming, Greedy and divide and conquer algorithm design strategies.

Problem 4

Describe a dynamic programming solution for 0-1 knapsack problem. Solve the following knapsack problems with dynamic programming approach just described.

a. n=4 items, capacity of knapsack M=8

Item (i)	Value (v_i)	Weight (w_i)
1	15	1
2	10	5
3	9	3
4	5	4

b. value = [20, 5, 10, 40, 15, 25]
weight = [1, 2, 3, 8, 7, 4]
capacity of knapsack M = 10

c. n=4 items, capacity of knapsack M=5

Item (i)	**Value (v_i)**	**Weight (w_i)**
1	3	2
2	4	3
3	5	4
4	6	5

Problem 5:

Describe the main ideas behind greedy algorithms with an example other than fractional knapsack problem. Give an example where the greedy approach does not yield an optimal solution.

Given a value V, to make change for V Rs, assuming infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 2000} valued coins/notes, one can find the minimum number of coins and/or notes needed to make the change.

For Examples:
If Input: V = 70
Output: 2 (a 50 Rs note and a 20 Rs note).

Input: V = 121
Output: 3 (a 100 Rs note, a 20 Rs note and a 1 Rs coin)

Write and describe a greedy algorithm to solve this problem. Also, explain the time complexity of your algorithm.

Note:- Typed solution (or clear image of handwritten solution) to this assignment may be mailed to suvendu.c@technoindiaeducation.com on or before 31/03/2020. However, one hardcopy of the solution need to be submitted in channel file, when the university reopens. Date of submission of the hardcopy should be announced later.