



AFIFA TAZREMIN OISHI

ID : 19-40028-1

SECTION : E

Date of submission :8th September ,2020

Algorithm Final Assignment Part – 1

Set 01

Answer to the question no: 1

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. Two properties that must be present in a problem to be approached by a dynamic programming that are:

- **Optimal Substructure:** If an optimal solution contains optimal sub solutions then a problem exhibits optimal substructure.
- **Overlapping subproblems:** When a recursive algorithm would visit the same subproblems repeatedly, then a problem has overlapping subproblems.

If a problem has optimal substructure, then we can recursively define an optimal solution. If a problem has overlapping subproblems, then we can improve on a recursive implementation by computing each subproblem only once.

If a problem doesn't have optimal substructure, there is no basis for defining a recursive algorithm to find the optimal solutions. If a problem doesn't have overlapping sub problems, we don't have anything to gain by using dynamic programming.

MCM means Matrix Chain Multiplication. The properties of MCM:

We can solve MCM in many ways - if the given chain is of 4 matrices. let the chain be ABCD, then there are 3 ways to place first set of parenthesis outer side:

(A)(BCD), (AB)(CD) and (ABC)(D). So when we place a set of parenthesis, we divide the problem into subproblems of smaller size.

Answer to the question no: 2

Reward Points = { 500, 210, 100, 700, 20 };

Price = { 5, 2, 4, 3, 1 };

Max price = 9;

Item = 5;

			0	1	2	3	4	5	6	7	8	9
Points	Price	Item	0	0	0	0	0	0	0	0	0	0
500	5	1	0	0	0	0	0	500	500	500	500	500
210	2	2	0	0	210	210	210	500	500	710	710	710
100	4	3	0	0	210	210	210	500	500	710	710	710
700	3	4	0	0	210	700	700	910	910	910	1200	1200
20	1	5	0	20	210	700	720	910	930	930	1200	1220

The maximum reward point is 1200 so Mr. Kerry should buy item 1,4 and 5 .

Answer to the question no: 3

I prefer 0 – 1 Knapsack algorithm to solve the above problem. The optimal solution to the 0-1 problem cannot be found with the same greedy strategy. So we can not use greedy method. Greedy method also gives us fractional number but we can not use any fractional number to solve the above problem. To make a greedy algorithm, identify an optimal substructure or subproblem in the

problem. Then, determine what the solution will include (for example, the largest sum, the shortest path, etc.). So we can not use greedy method.

To solve the above problem we can not also use brute-force method because brute force method has no preprocessing phase, it is lengthy process, takes many time and high cost, need extra space again it will give optimal solution with poor efficiency.

-----O-----