Page No.
Date

Assignment 3

Fill in the blanks

1) Bellmann Ford Algorithm can be applied for weighted Graph Data Structure.

2) Floyd Warshall's Algorithm is used for solving all pairs shortest path problem.

3) If a problem can be solved by combining optimal solutions to non-overlapping problems, the strategy is called divide and conquer.

Choose Correct Options

1) Time complexity of fractional knapsack problem is

a) O(n log n)

2) If an optimal solution can be created for a problem by constructing optimal solutions for its subproblems, the problem possesses

a) Optimal substructure.

3) Which of the following problems should be solved using dynamic programming?

a) Longest common subsequence

4) When a top-down approach of dynamic

programming is applied to a problem, it

Ans: b) Decreases the time complexity and increases space complexity.

c) Which of the following is/are property/pr of a dynamic programming problem?

Ans: d) Both optimal substructure and overlapping subp

Q3) State whether the following statements are tr false

a) When dynamic programming is applied to a p it takes far less time as compared to other that don't take advantage of overlapping su True.

b) A greedy algorithm can be used to solve all dynamic programming problems. False.

c) Fractional knapsack problem can be solv time O(n). True.

Q4) Name the following or define or design the follo

a) What is dynamic programming?

Ans: Dynamic programming is an algorithm paradigm majorly used to formulate solutions to the optimization problems.

How does a greedy algorithm contruct the solution?

To construct the solution in an optimal way, this algoritum creates two sets where one set contains all the chosen items, and another set contains the rejected items.

What is greedy algorithm?

Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious & immediate benefit.

Answer the following questions in brief (20 to 3 words)

What are the characteristics of dynamic programming?

The characteristics of dynamic programming are:
It breaks down the complex problem into simpler subproblems.
It finds the optimal solution to these sub-problems.
It stores the results of subproblems.
It reuses them so that same subproblem is calculated more than once.

Explain 0/1 Knapsack Problem.

The 0/1 knapsack problem means that the items are either completely or no items are filled in

a knapsack. For eg, we have 2 kg & 3 if we pick the 2kg item then we cannot 1kg item the 2kg item, we have to the 2kg item completely.

c) Explain Multistage graphs.

Ans: A multistage graph is a directed graph in the nodes can be divided into a set of such that all edges are from a stage to next stage only.

Q6) Answer the following questions in brief (50 words)

a) Obtain the solution to knapsack problem greedy method $n=7$, $m=15$ $(p_1, p_2 \dots p_7) = (15, 7, 6, 18, 3)$, $(w_1, w_2, \dots, w_7) = (2, 3, 5, 7,$

Ans: Given: $n=4$, $m=8$, $(p_1, p_2 \dots p_4) = (3, 5, 6, 10)$
$(w_1, w_2, \dots, w_4) = (2, 3, 4, 5)$
To prove: Optimal solution that gives maxi profit.
Proof:
Step 1: (To find profit / weight ratio)
$p_1/w_1 = 10/2 = 5$
$p_2/w_2 = 5/3 = 1.67$
$p_3/w_3 = 15/5 = 3$
$p_4/w_4 = 7/7 = 1$
$p_5/w_5 = 6/1 = 6$
$p_6/w_6 = 18/4 = 4.5$

p7/w7 = 3/1 = 3

Arrange this profit/weight ratio in non-increasing order as n value) Since the highest profit/weight ratio is 6. That is p5/w5, so 1st value is 5. Second highest profit/weight ratio is 5. That is p1/w1, so 2nd value is 1. Similarly, calculate such n values and arrange them in non-increasing order.

Order = (5, 1, 6, 3, 7, 2, 4)

Q3: (To find optimal solution using m = 15 & n = 7)

Consider x5 = 1, profit = 6

Then consider x1 = 1, profit = 10

So weight uptil now = 1+2 = 3

Now x6 = 1, profit = 18

So, total profit = 16+18 = 34

And weight uptil now = 3+4 = 7

Now x3 = 1, profit = 15

So total profit = 34 + 15 = 49

And weight uptil now = 7+5 = 12

Now x7 = 1, profit = 3

So total profit = 49+3 = 52

And weight uptil now = 12+1 = 13

∵ m = 15 so we require only 2 units more.

∴ x2 = 2/3

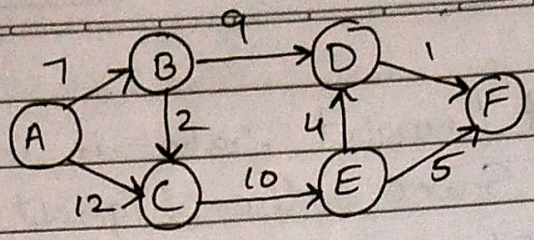
So total profit = 52 + 5 x 2/3 = 52 + 3.33 = 55.3

And weight uptil now = 13 + 3 x 2/3 = 15

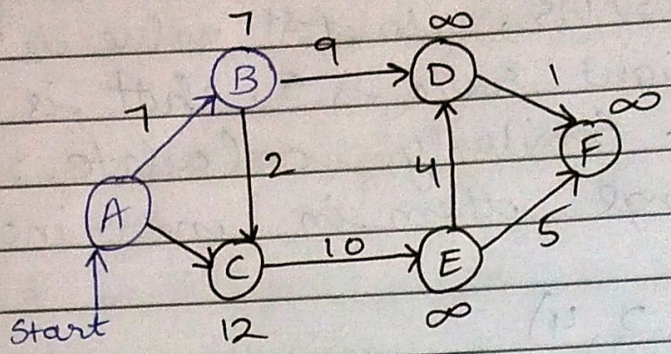
Thus, the optimal solution that gives maximum profit is, (1, 2/3, 1, 0, 1, 1, 1).

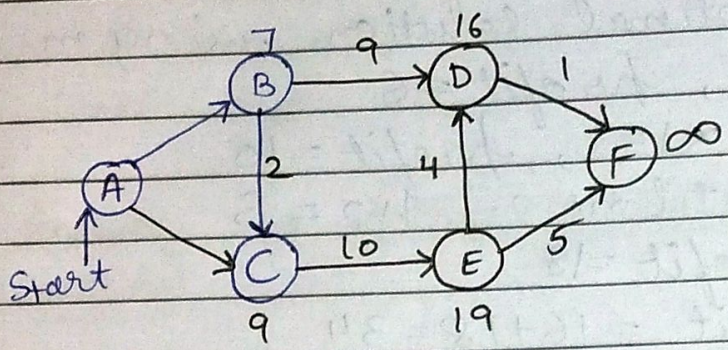
Apply Dijkstra's algorithm on given graph.

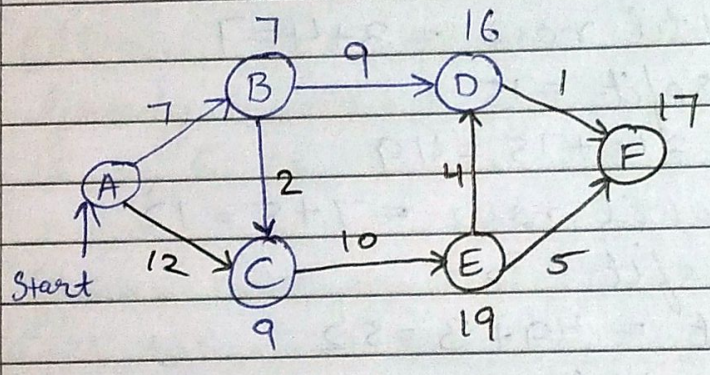
Page No.
Date

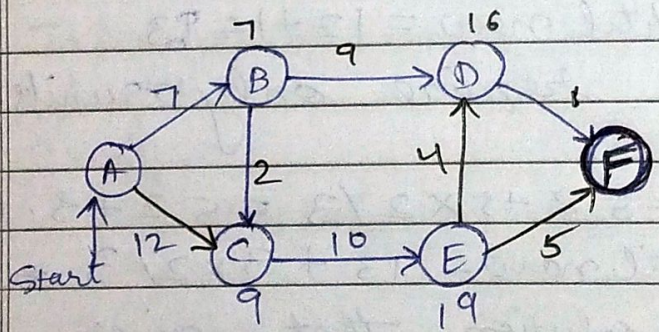


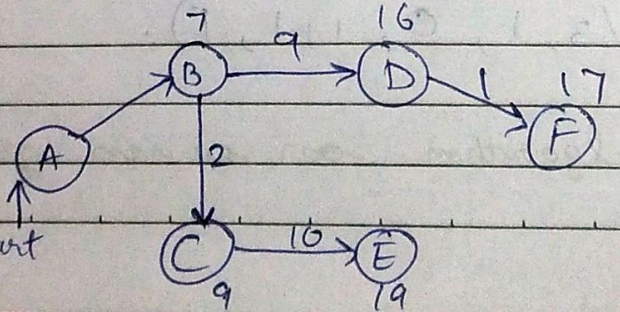
Ans':











Page No.
Date

Think and Answer

What are the differences between the dynamic programming and greedy approach?

Dynamic programming	Greedy method
In Dynamic programming we make decision at each step considering current problem & solution to previously solved subproblem to calculate optimal solution.	In a greedy algorithm, we make whatever choice seems best at the moment in the hope that it will lead to global optimal solution.
It is guaranteed that Dynamic Programming will generate an optimal solution as it generally considers all possible cases & then choose the best.	In greedy method, sometimes there is no such guarantee of getting optimal solution.
A Dynamic programming is an algorithmic technique which is usually based on a recurrent formula that uses some previously calculated states.	A greedy method follows the problem solving heuristic of making the locally optimal choice at each stage.

b) Explain Travelling Salesperson problem with examp

Ans: The travelling salesman problems abide by a sol
and a set of cities. The salesman has to
every one of the cities starting from certain
and to return to the same city. The
challenge of the problem is that the trave
salesman needs to minimize the total leng
of the trip.

Example: A newspaper agent daily drops the newspaper
to the area assigned in such a manner
he has to cover all the houses in the res
area with minimum travel cost. Find Co
the minimum travel cost.

c) Explain Assembly-line scheduling with example.

Ans: A car factory has two assembly lines, ea
with n stations. A station is denoted by S
where i is either 1 or 2 and indicates t
assembly line the station is on, and j is
the number of the station. The time taken
station is denoted by $a_{i,j}$. Each station
dedicated to some sort of work like eng
fitting, body fitting, painting and so on.
car chassis must pass through each
n stations in order before exiting th
factory. The parallel stations of the tw
assembly lines perform the same task. I
passes through station $S_{i,j}$, it will continue
station $S_{i,j+1}$ unless it decides to transfer
other line.