7-05-20

DAA - Revision

JOVIAL JOE
1ESI7CS016

Q.1) ⓐ Discuss about the fractional and 0/1 knapsack problem.

A1) ⓐ Knapsack Problem

- It is an optimization problem when considered from an computer science perspective.
- The problem goes like this:

"A hiker tries to pack the most valuable items without overloading the knapsack. Each item has a certain value/profit/benefit and weight. An overall weight limitation gives the single constrain"

Fractional Knapsack

- In this case, the item can be filled into the knapsack even if it ~~had~~ only a fraction of it would go into the knapsack.
- Mainly, three approaches are used to solve this problem, one moving from highest profit to lowest.

- Second, moving from least weight to maximum weight.
- Finally from maximum profit/weight ratio to the lowest of the same.

0-1 Knapsack Problem

- In this type of problem, ~~people are~~ hiker is only allowed to take items as a whole.
- If an item ~~can~~ cannot be taken as a whole, it must be left behind.
- Parts of items cannot be taken.

Q 1 (b) Calculate the knapsack solution, where $j = 3$, $m = 30$ whieghts are (10, 12, 15) and profits are (20, 28, 22)

Sol

Using 0-1 Knapsack Method

Given: $j = 3$

$m = 30$

weights = 10, 12, 15

profit = 20 28 22

$$\sum_{j=1}^{3} P_i x_i \text{ = net profit}$$

$(x_i = 0 \text{ or } 1)$

(i) Selection of object with largest profit:

A (28, 12) C (22, 15) B (20, 10)

selected	profit	remain-weigh
A	28	18
B	22	3

$\Rightarrow$ Profit $= 28 \times 1 + 22 \times 1 = 50$

(ii) Selection of object with smallest weigth.

B (20, 10), A (28, 12), C (22, 15)

Selected	Profit	Remaining weight
B	20	20
A	28	8

$\rightarrow$ Profit = $20 \times 1 + 28 \times 1 + 22 \times 0 = \underline{\underline{40}}$

$\therefore$ Use maximum profit method.

Q 1 c)

Greedy Method	Dynamic Programming
~ Greedy algorithm makes whatever choice that seems best at the moment.	~ Decisions are made considering current ~~and previous~~ problem & solution to previously solved subprobem
~ No guarantee of getting an optimal solution.	~ Will produce an optimal solution for sure
~ It is storage efficient	~ Increses in space complexity.
~ General faster - in the order of logn.	~ Genally slower - in the order of n^2.
eg.- fractional Knapsack (without ordering)	eg.- 0/1 knapsack problem.

Using fractional - knapsack

1.) Maximum Profit First:

Selected	Profit	Remaining
A	28	18
C	22	3
$\frac{3}{10}$ B	$\frac{3}{10} \times 20$	0

$\Rightarrow$ Net profit $= 28 + 22 + 6 = 56.00$

2.) Minimum Weight First

Selected	Profit	Remaining
B	20	20
A	28	8
$\frac{8}{15}$ C	$\frac{8}{15} \times 22$	0

Net profit $= 20 + 28 + \frac{8}{15} \times 22 = 59.73$

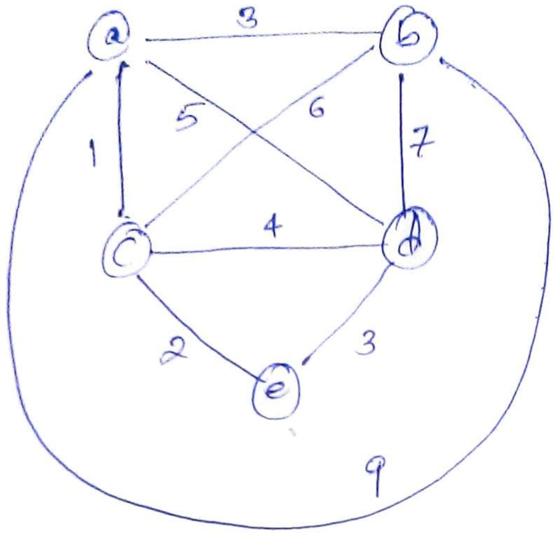
3.) Maximum profit/weigth ratio first

A (P/W) = 2.33 B (P/W) = 2.00 C (P/W) = 1.466

Selected	Profit	Remaining
A	28	18
B	20	8
$\frac{8}{15}$ C	$\frac{8}{15} \times 22$	0

Net profit $= 28 + 20 + \frac{8}{15} \times 22 = \underline{\underline{59.73}}$

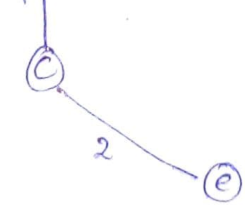
Q 2(a) Construct a minimum spanning tree for the below graph using prim's algorithm. (source node = a)

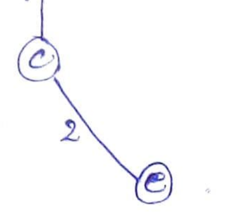


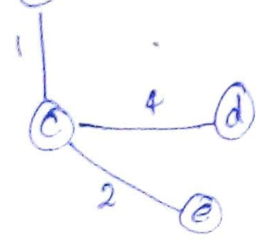
Using Prim's Algorithm

(1)

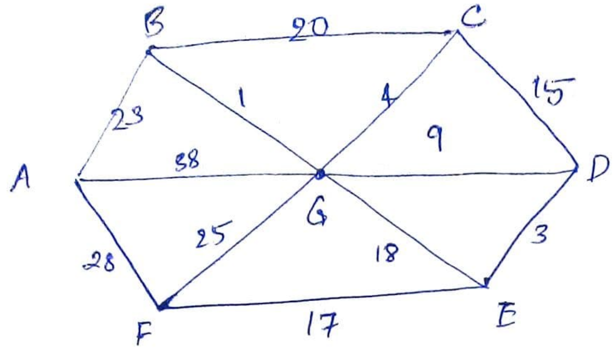
(2) 

(3) 

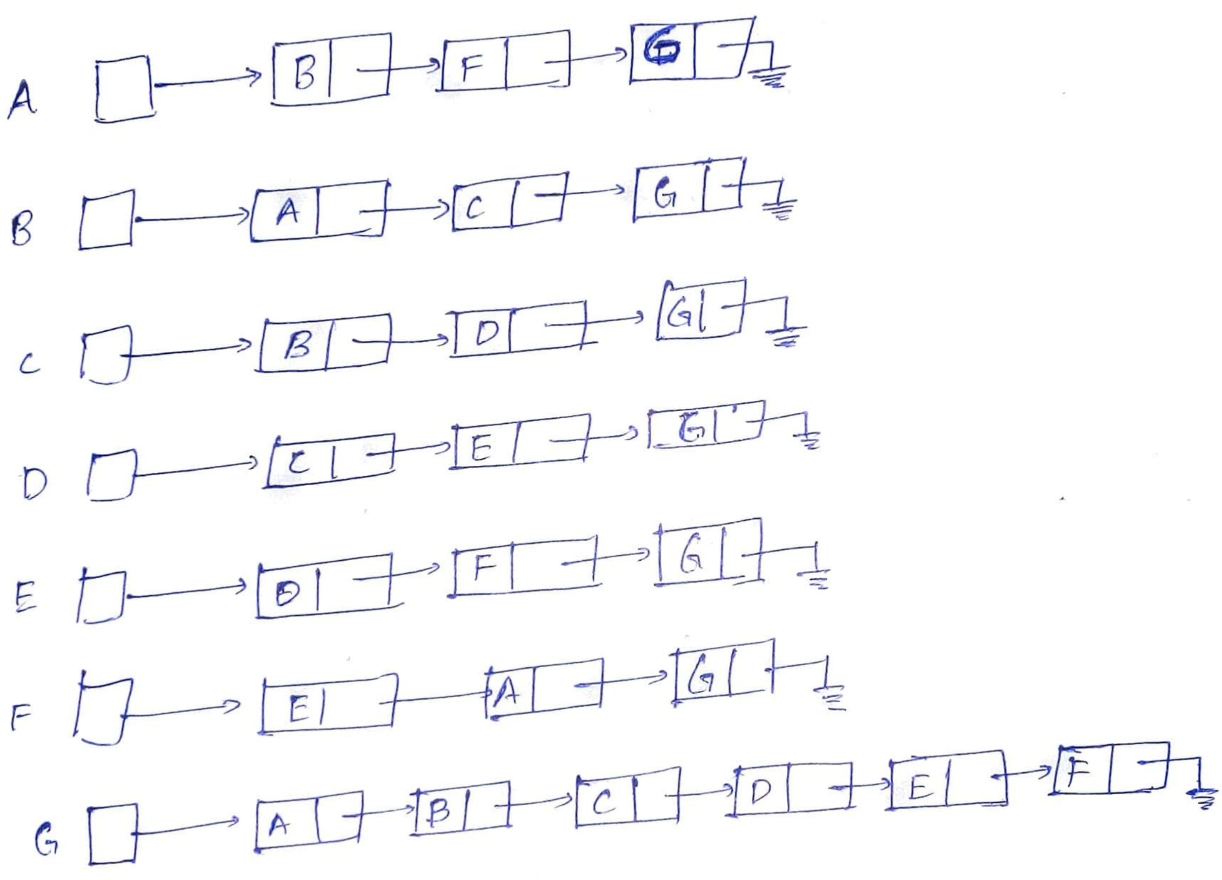
(4) 

(5) 

8 (b).)



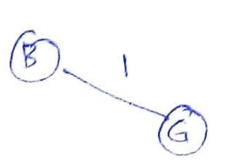
(i) Adjacency list representation of the graph.



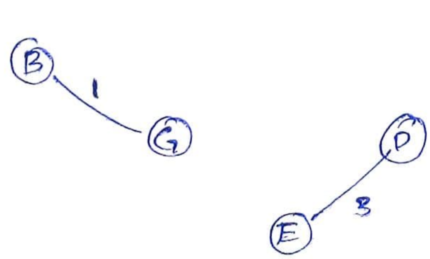
This is the adjacency list repestation of the graph.

(ii) MST using krus kals algorithm

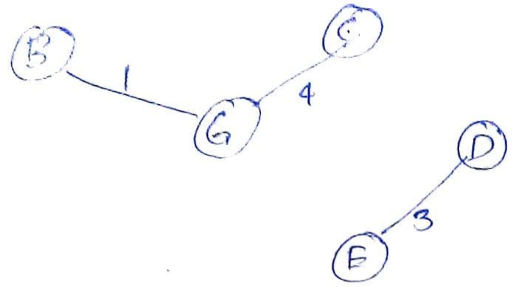
(1)



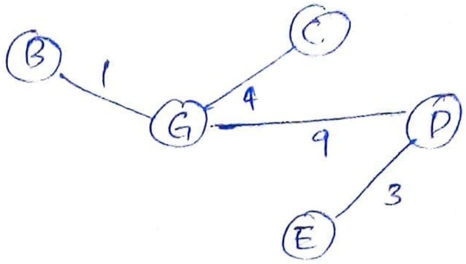
(2)



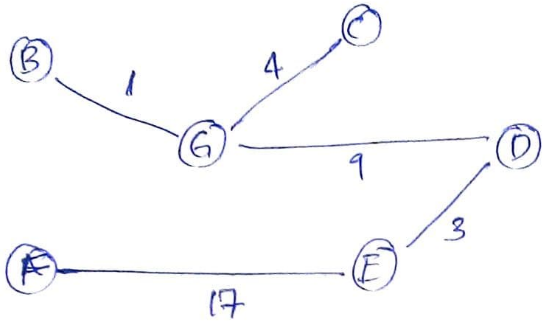
(3)



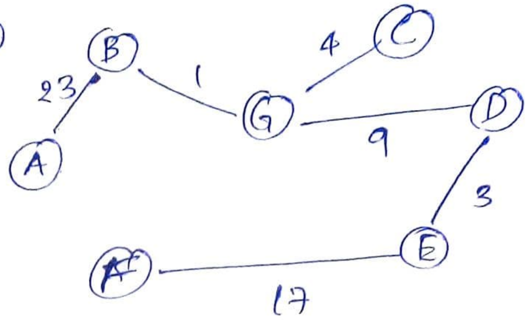
(4)



(5)



(6)



This is the required MST.