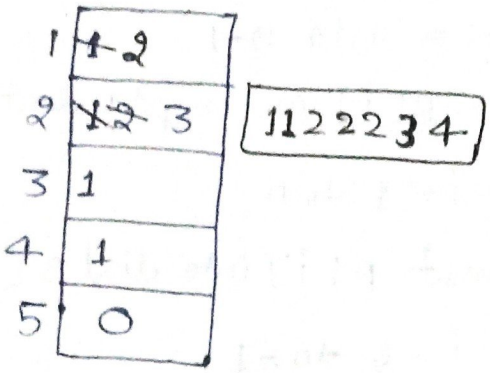
- DAA

① Counting Sort

* Given Input size is "n"
* Given Range is "K"
* it is not comparision based sorting

(1-5) value - 2 1 2 3 1 2 4



② Radix Sort ⇒

it is non comparision based algoritm.

Q.) 904, 46, 5, 74, 62, 1

904, 046, 005, 074, 062, 001 +जो नम्बर सबसे बड़ा होगा उसी के बराबर सब को कर लेंगे

Sort ⇒ last digit ⇒

001
062
904
074
005
046

Sort (10Place) ⇒

001
904
-005
-046
-062
-074

001
005
046
062
074
904

finally Sort value

③ Bucket Sort ⇒ non comparision based algorithm.

Bucket – Sort (A)

1) Id B(0...n-1) be a new array.
2) n = A. length
3) for i = 0 to n-1
4) make B[i] an empty list.
5) for i = 1 to n
6) insert A[i] into list B[[nA[i]]]
7) for i = 0 to -1.
8) Sort B[i] with insertion sort.
9) concatenate the lists together B[0]. B[1] B[n]
in order.

Q.1) 0.79, 0.13, 0.64, 0.39, 0.20, 0.89, 0.53, 0.42, 0.06, 0.9

0	.06
1	.13
2	.20
3	.39
4	.42
5	.53
6	.64
7	.79
8	.89
9	.94

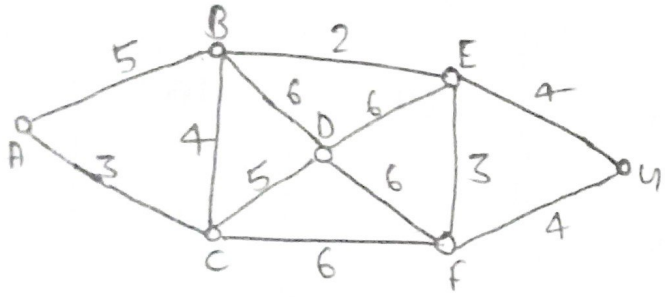
*दशमलव के बाद जो Number होगा उसी के हिसाब से लिखेंगे।

0.06	0.13	0.20	0.39	0.42	0.53	0.64	0.79	0.89	0.94

⇓

This Value Sort

Minimum Spanning tree



① constrant min heap with 'e' edges

② Take one by one edge and add in spanning tree (cycle should be not created)

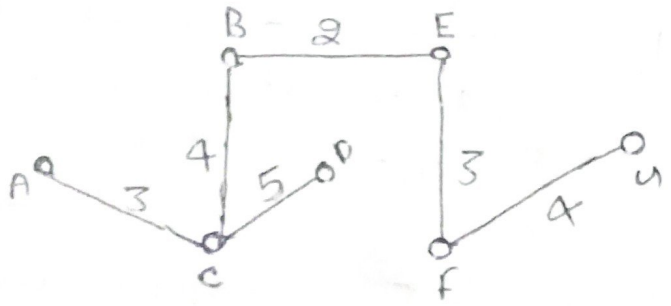
Best case (n-1) edges
worst case (e) edges

* Spanning tree बनाते समय cycling नहीं होनी चाहिए)

* सबसे पहले हम minimum value से start करेंगे

Solve

v → edge
(n-1)
7 - 1 = (6)



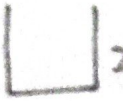
Total cost of minimum tree = (21)

Fractional KnapSack Problem

(1.)

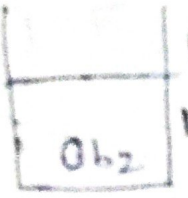
Object	Ob1	Ob2	Ob3
Profit	25	24	15
weight	18	15	10
	1.3	1.6	1.5

Knapsack Capacity (m) = (20)



Solve

P/w Bath



5 + जो सबसे 15. 24 इसी Object होगा उसी की लिखो

$\frac{5 \times 15}{10} = 7.5$

24 + 7.5 = (31.5)

Ans

: for i = 1 to n

→ Calculate P/w

→ Sort object in decreasing order of P/w Ratio

→ for i = 1 to n

if m > 0 and wi ≤ m

m = m − wi,

P = P + Pi,

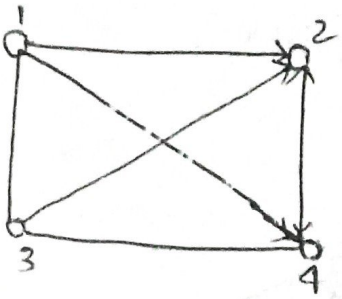
else break

If (m > 0)

P = P + Pi (m/wi)

Travelling Salesman Problem

Q.)



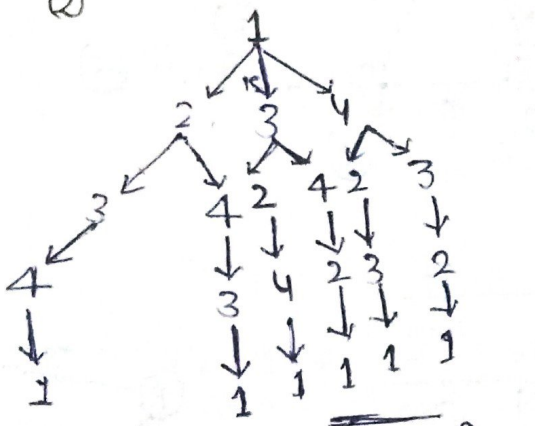
	1	2	3	4
1	0	10	15	20
2	5	0	25	10
3	15	30	0	5
4	15	10	20	0

Solve * हम minimum Value से शुरू करेंगे हमेशा

① Greedy method

1 → 10
2 → 10
4 → 20
3 → 15
1
55

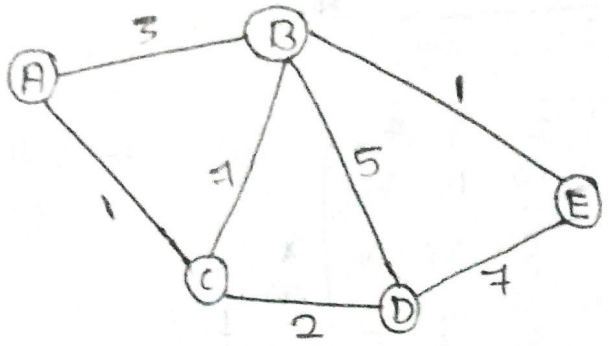
②



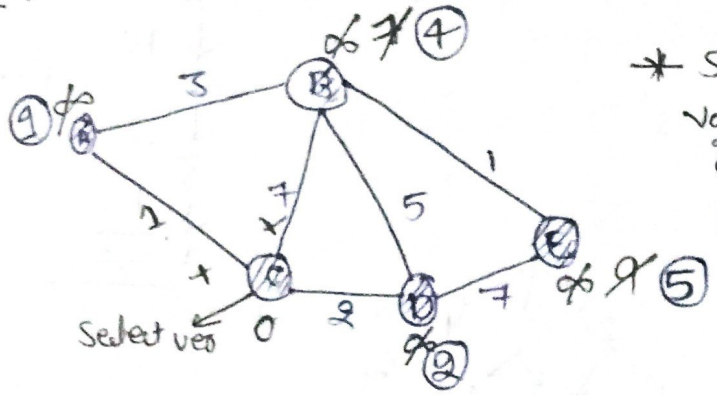
35 Ans

Dijkstras Algorithm

Q) find the Soodest path in the given weight graph by Dijkstoa algorithm.

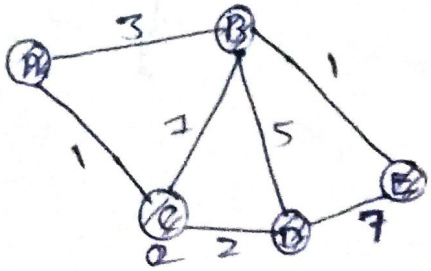


Solve ⇒



* Select करते समय vertices को हमेशा minimum लेंगे

Select vertex	A	B	D	E
A	(1) ←	7	2	∞
D		4	(2)	∞
B		(4) ←		9
				(5)



C A — CA → C – A

C B — CAB → C – A – B

C D — CD → C – D

C E — CABE → C – A – B – E