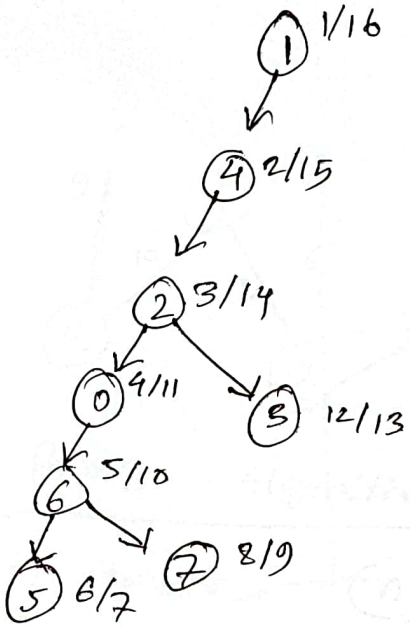
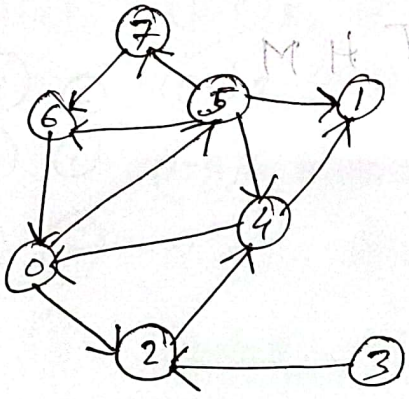
Ans: to the Question NO-01

Strongly Connected Components

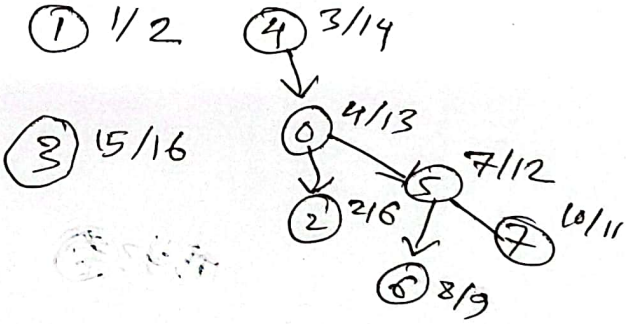
- DFS on the given graph



- Reversing the edge



- DFS on the reverse graph



- Strongly Connected components: (i) (1)

(ii) 0, 2, 4, 5, 6, 7

(iii) (3)

(2) Ans: to the question: 02

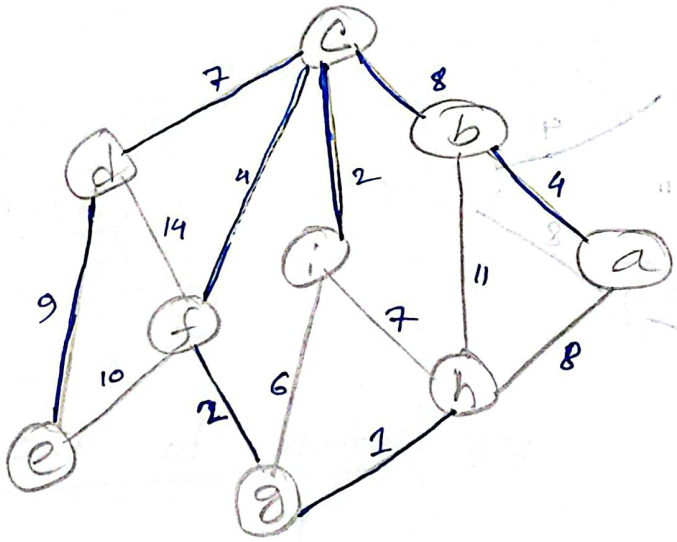
	Ø	A	L	G	O	R	I	T	H	M
Ø	0	0	0	0	0	0	0	0	0	0
L	0	0	1	1 →	1 →	1 →	1 →	1 →	1 →	1
I	0	0	1 ↓	1 →	1 →	1 →	2 →	2 →	2 →	2
T	0	0	1 ↓	1 →	1 →	1 →	2 ↓	3 →	3 →	3
H	0	0	1 ↓	1 →	1 →	1 →	2 ↓	3 ↓	4 →	4
I	0	0	1 ↓	1 →	1 →	1 →	2 ↓	3 ↓	4 ↓	4
U	0	0	1 ↓	1 →	1 →	1 →	2 ↓	3 ↓	4 ↓	4
M	0	0	1 ↓	1 →	1 →	1 →	2 ↓	3 ↓	4 ↓	5

Length of LCS = 5

The LCS = L I T H M

Ans: to the question no-03

(h)

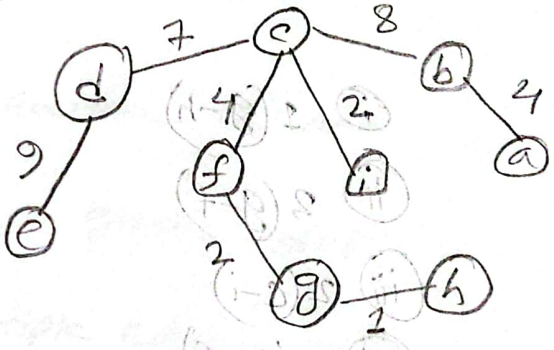


Prim's Algorithm

We iterate (n-1) or (9-1) times
= 8 times

(i) (a) (a-b) 4, (a-h) 8
(ii) (b) (b-c) 8, (b-h) 11
(iii) (c) (c-i) 2, (c-f) 4, (c-d) 7
(iv) (i) (i-g) 6, (i-h) 7
(v) (f) (f-g) 2, (f-e) 10, (f-d) 14
(vi) (g) (g-i) 6, (g-h) 1
(vii) (h) (h-a) 8, (h-i) 7, (h-b) 11
(viii) (d) (d-e) 9, (d-f) 14

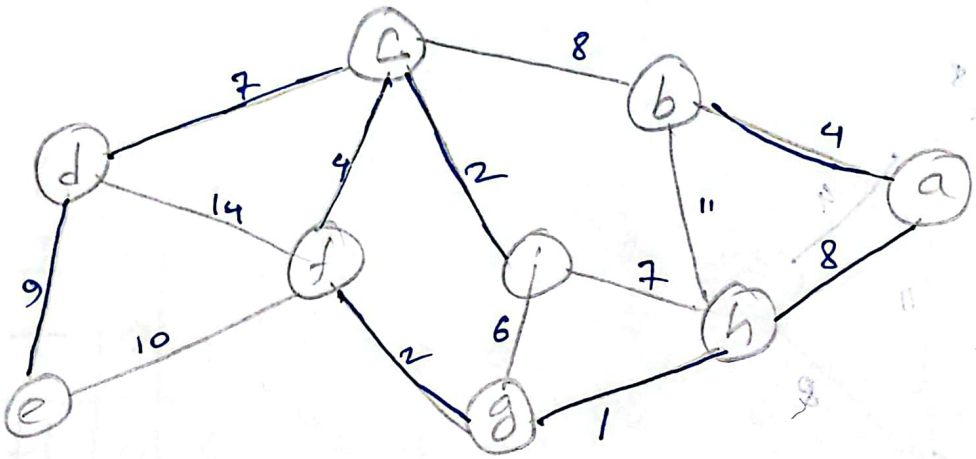
Required MST



Cost of MST : 37

(4)

Ans: to the Q: No-of 4



Kruskal's Algorithm

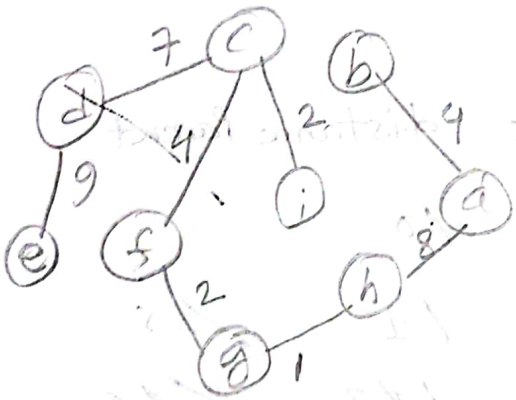
We iterate $(n-1)$ or $(9-1)$ times
$= 8$ times.

(i) 1 (g-h)
(ii) 2 (g-f)
(iii) 2 (c-i)
(iv) 4 (c-f)
(v) 4 (b-a)
(vi) 7 (c-d)
(vii) 8 (a-h)
(viii) 9 (d-e)

* (g-i), (i-h) can't be taken because these vertices form a cycle.

(b)

Required MST

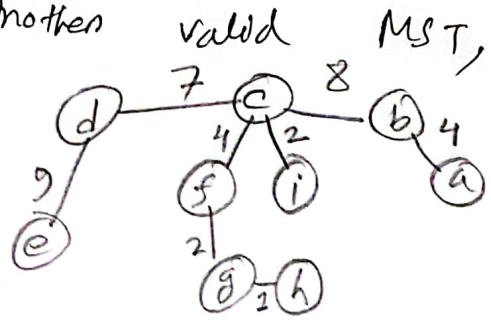


cost of MST : 37

Ans: to the Q: NO -05

(i) Minimum spanning tree of question (3) and (4) are different. Because the given graph in Figure- Graph-1 has mutiple edge with the same weight, i.e :- (h-a) → 8
(b-c) → 8

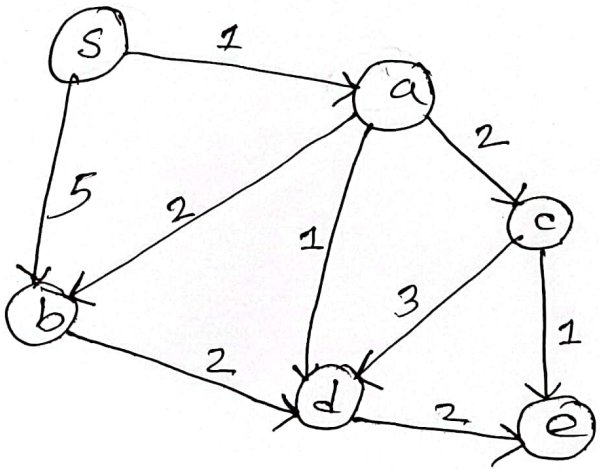
Another valid MST,



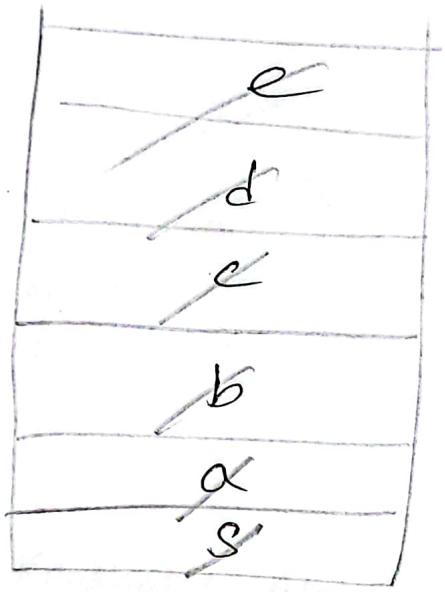
In this case, if I choose edge (b-c) instead of (h-a) then both MSTs become the same.

(ii) The cost of the both Minimum Spanning Tree are same.

Ans: to the Question NO-06



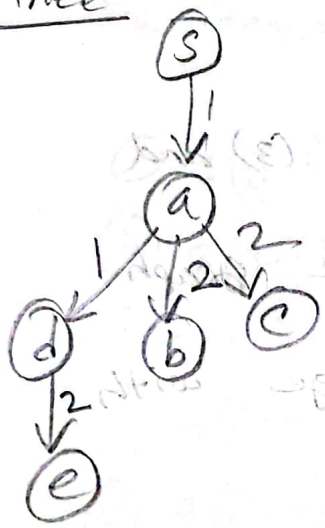
(7)



Priority queue

Vertex	distance	Parent
s	~~∞~~ 0	NIL
a	~~∞~~ 1	~~NIL~~ s
b	~~∞~~ ~~3~~ 3	~~NIL~~ ~~a~~ a
c	~~∞~~ 3	~~NIL~~ a
d	~~∞~~ 2	~~NIL~~ a
e	~~∞~~ 4	~~NIL~~ d

Tree



Shortest Path from source

(s-a) : s → a cost: 1

(s-c) : s → a → c cost: 3

(s-b) : s → a → b cost: 3

(s-d) : s → a → d cost: 2

(s-e) : s → a → d → e cost: 4