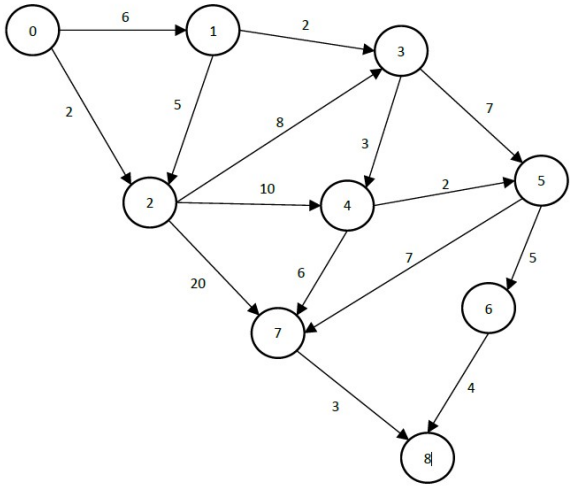
ASSIGNMENT-3

Shortest path algorithm: Label correction method.
Home tree and guest node concept used.

Q.1:



Step-1 (node-0)
Home tree – {0}, d(0) = 0
Guest node - {1,2,3,4,5,6,7,8}, d(guest nodes) - ∞
Distance 0-1 = 6
Distance 0-2 = 2
Updated: d(1) = 6, d(2)=2

Step-2 (node-2)
Home tree – {0,2}, d(0) = 0, d(1)=6, d(2)=2
Guest node - {1,3,4,5,6,7,8}, d(gn) - ∞
Distance 2-3 = 2+8 = 10
Distance 2-4 = 2+10 =12
Distance 2-7 = 2+20 =22
Updated: d(3) = 10, d(4)=12, d(7)=22

Step-3 (node-1)
Home tree – {0,1}, d(0) = 0, d(1)=6, d(2)=2, d(3)=10, d(4)=12, d(7)=22
Guest node - {2,3,4,5,6,7,8}, d(gn) - ∞
Distance 1-2 = 5+6=11
Distance 1-3 = 2+6=8
Updated: d(2) = 2, d(3)=8

Step-4 (node-3)
Home tree – {0,1,2,3}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=12, d(7)=22
Guest node - {4,5,6,7,8}, d(gn) - ∞
Distance 3-4 = 3+8 =11
Distance 3-5 = 7+8 =15
Updated: d(4)=11, d(5)=15

Step-5 (node-4)
Home tree – {0,1,2,3,4}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=11, d(7)=22, d(5)=15
Guest node - {5,6,7,8}, d(6,8) - ∞
Distance 4-5 = 2+11=13
Distance 4-7 = 6+11 =17
Updated: d(5)=13, d(7)=17

Step-6 (node-7)
Home tree – {0,1,2,3,4,7}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=11, d(7)=17, d(5)=13
Guest node - {5,6,8}, d(6,8) - ∞
Distance 7-8 = 3+17=20
Updated: d(8)=20

Step-7 (node-5)
Home tree – {0,1,2,3,4,5}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=11, d(7)=17, d(5)=13, d(8)=20
Guest node - {7,6,8}, d(6) - ∞
Distance 5-7 = 7+13=20
Distance 5-6 = 5+13=18
Updated: d(7)=17, d(6)=18

Step-8 (node-6)
Home tree – {0,1,2,3,4,5,6,7}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=11, d(7)=17, d(5)=13, d(8)=20, d(6)=18
Guest node - {8}
Distance 6-8 = 4+18=22
Updated: d(8)=20

Step-9 (node-8)
Home tree – {0,1,2,3,4,5,6,7,8}, d(0) = 0, d(1)=6, d(2)=2, d(3)=8, d(4)=11, d(7)=17, d(5)=13, d(8)=20, d(6)=18
Guest node - {}

Label corrected

Shortest path is **0-1-3-4-7-8**

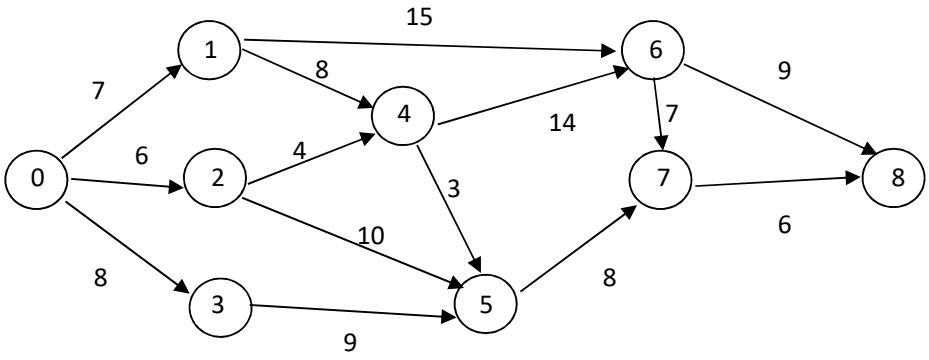
Q.2:

Python output.

Agent_id: 16 from node 0 to node 8 shows that the shortest path from node 0-8 is **0-1-3-4-7-8**

agent_id	from_node	to_node	shortest_path_cost	shortest_path_node_seq
1	0	1	6	0;1;
2	0	2	2	0;2;
3	1	3	2	1;3;
4	1	2	5	1;2;
5	2	3	8	2;3;
6	2	4	10	2;4;
7	2	7	16	2;4;7;
8	3	4	3	3;4;
9	3	5	5	3;4;5;
10	4	5	2	4;5;
11	4	7	6	4;7;
12	5	6	5	5;6;
13	5	7	7	5;7;
14	6	8	4	6;8;
15	7	8	3	7;8;
16	0	8	20	0;1;3;4;7;8;

Q.3:



agent_id	from_node	to_node	shortest_path_cost	shortest_path_node_seq
1	0	1	7	0;1;
2	0	2	6	0;2;
3	0	3	8	0;3;
4	1	6	15	1;6;
5	1	4	8	1;4;
6	2	4	4	2;4;
7	2	5	7	2;4;5;
8	3	5	9	3;5;
9	4	6	14	4;6;
10	4	5	3	4;5;
11	5	7	8	5;7;
12	6	7	7	6;7;
13	6	8	9	6;8;
14	7	8	6	7;8;
15	0	8	27	0;2;4;5;7;8;

Python output.

Agent_id: 15 from node 0 to node 8 shows that the shortest path from node 0-8 is **0-2-4-5-7-8**