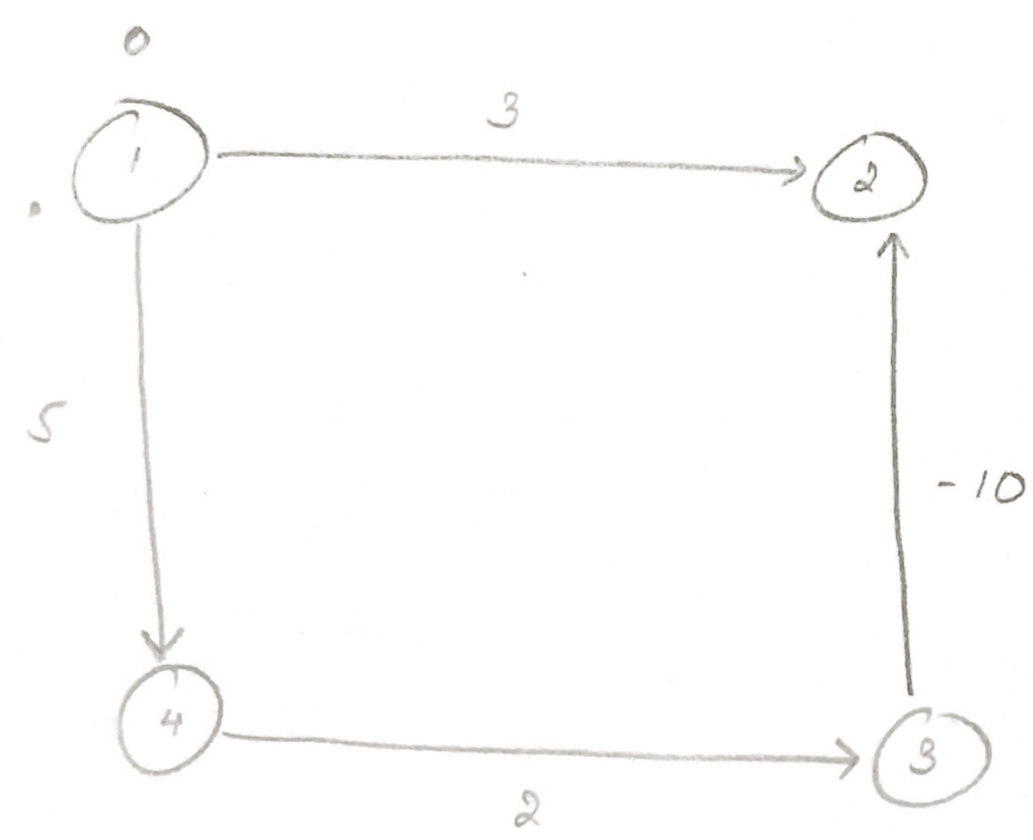
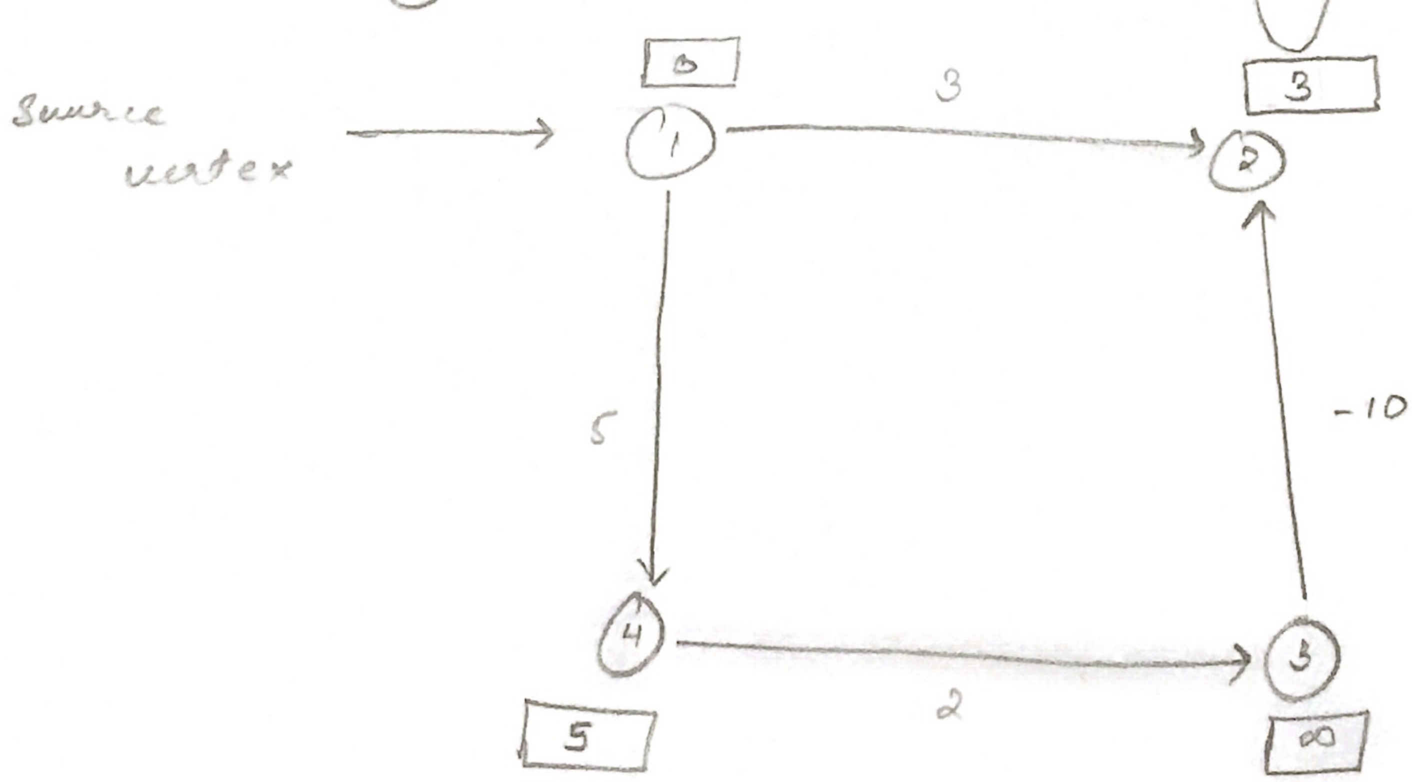
Q.5. (a) Dijkstra Algorithm with negative weight example:



Considering the above example we see that path from (3, 2) has a negative weight which is -10.

(a) Now lets label vertices with weight/Cost. by selecting vertex (1)



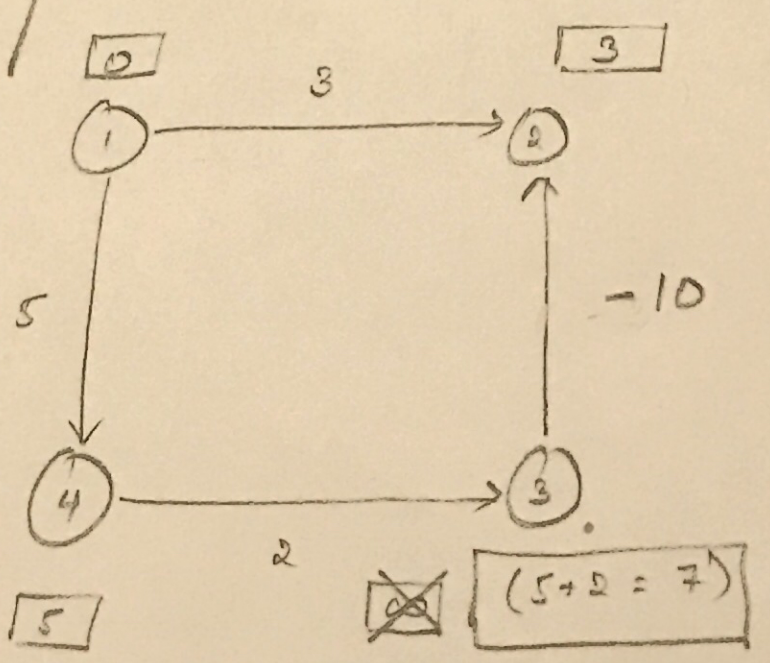
We label vertex 2 and 4 with 3 and 5 respectively because source has direct edge to these vertices. And that there is no direct edge path from (1,3) therefore vertex 3 has been labeled as ∞. Vertex 1 has been labeled 0 because it is the source vertex.

Now, this is the initial setup of Dijkstra.

(b) Now, the next shortest path is with vertex 2 with cost 3. There is no connected vertex to this one. Therefore none other vertex can be relaxed because there is no outgoing edge. Thus, we move on.

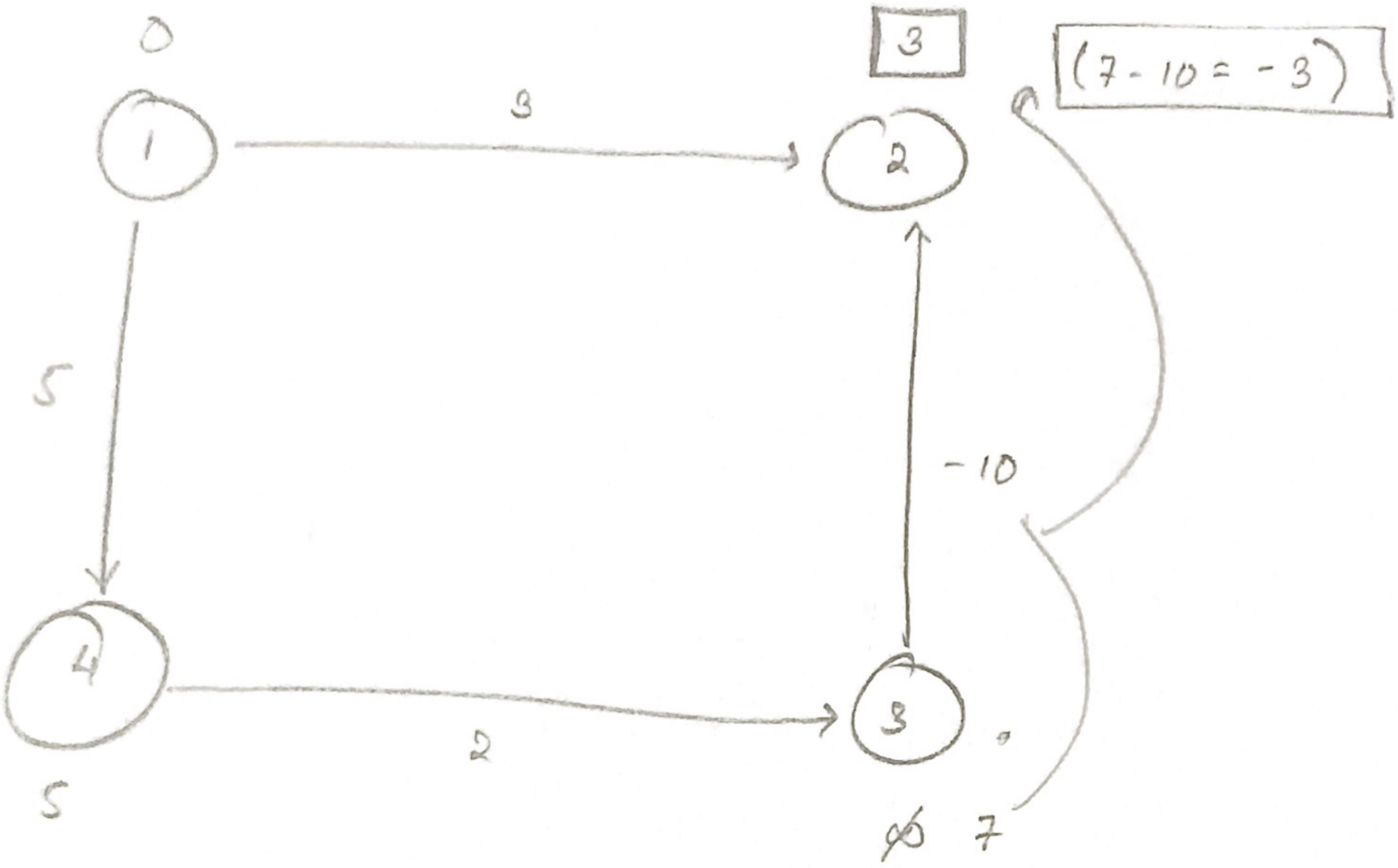
(c) Select the the next vertex with smallest cost. Here, which is 4. Vertex 4 will relax vertex 3 because there is an outgoing edge.

Therefore graph looks like this:



Hence vertex 3 has cost of 7.

(d) Now that vertex (2) is already relaxed lets go back to check once.



Going back and checking (-3) seems to be a better answer but initially setting up the problem using Dijkstra algorithm approach we get 3

∴ In this case Dijkstra algorithm greedy approach does not work with the forwarded negative edge.

<u>Hence proved.</u>

Also order of selection: (1), (2), (4), (3) for vertex

(a) Let $G = \{E, V\}$ be weighted and directed graph.

Consider the following graph and solve using Dijkstra algorithm.

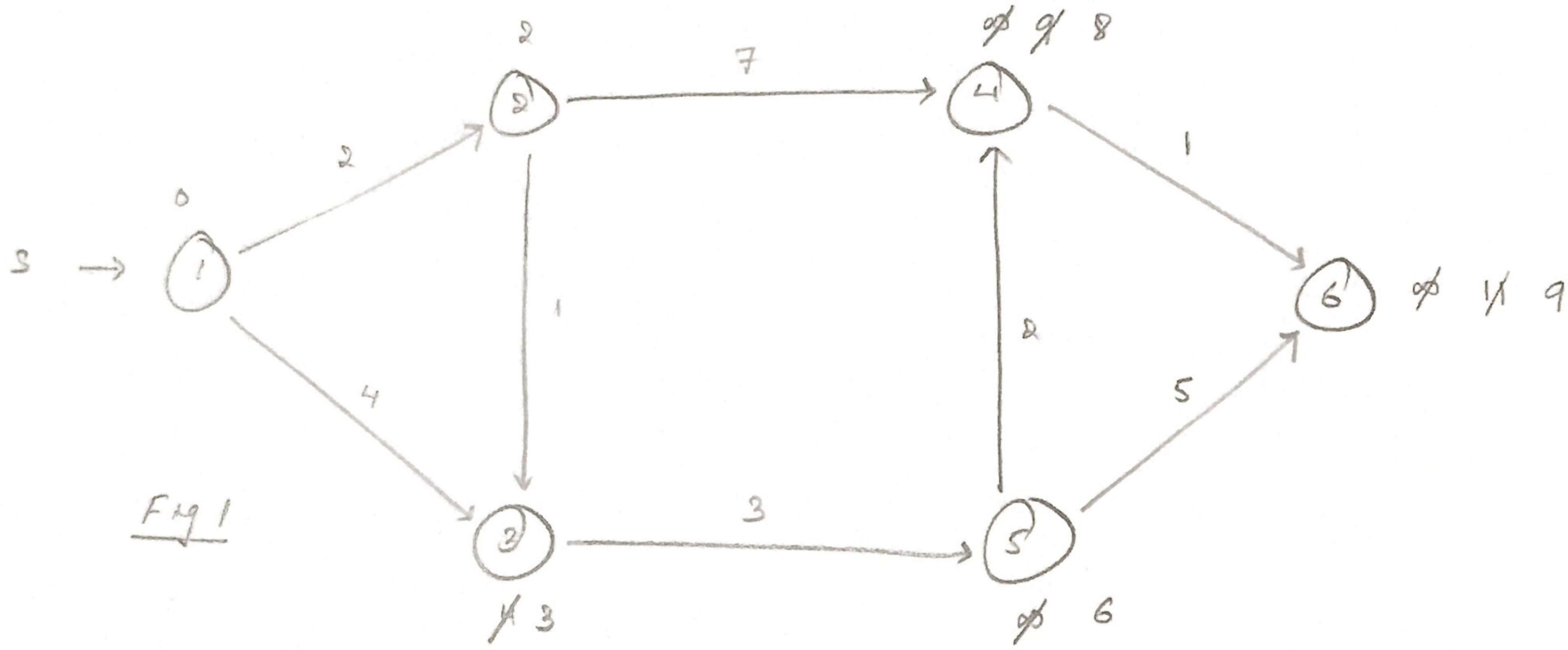


Fig 1

S	V	d [V]
1	2	2
1	3	3
1	4	8
1	5	6
1	6	9

Where V is vertices and d [V] is the distance to all the vertices and S is the source.

Now if we increment the weight of an edge by a constant c.

let that constant be 5. Then building the graph we get.

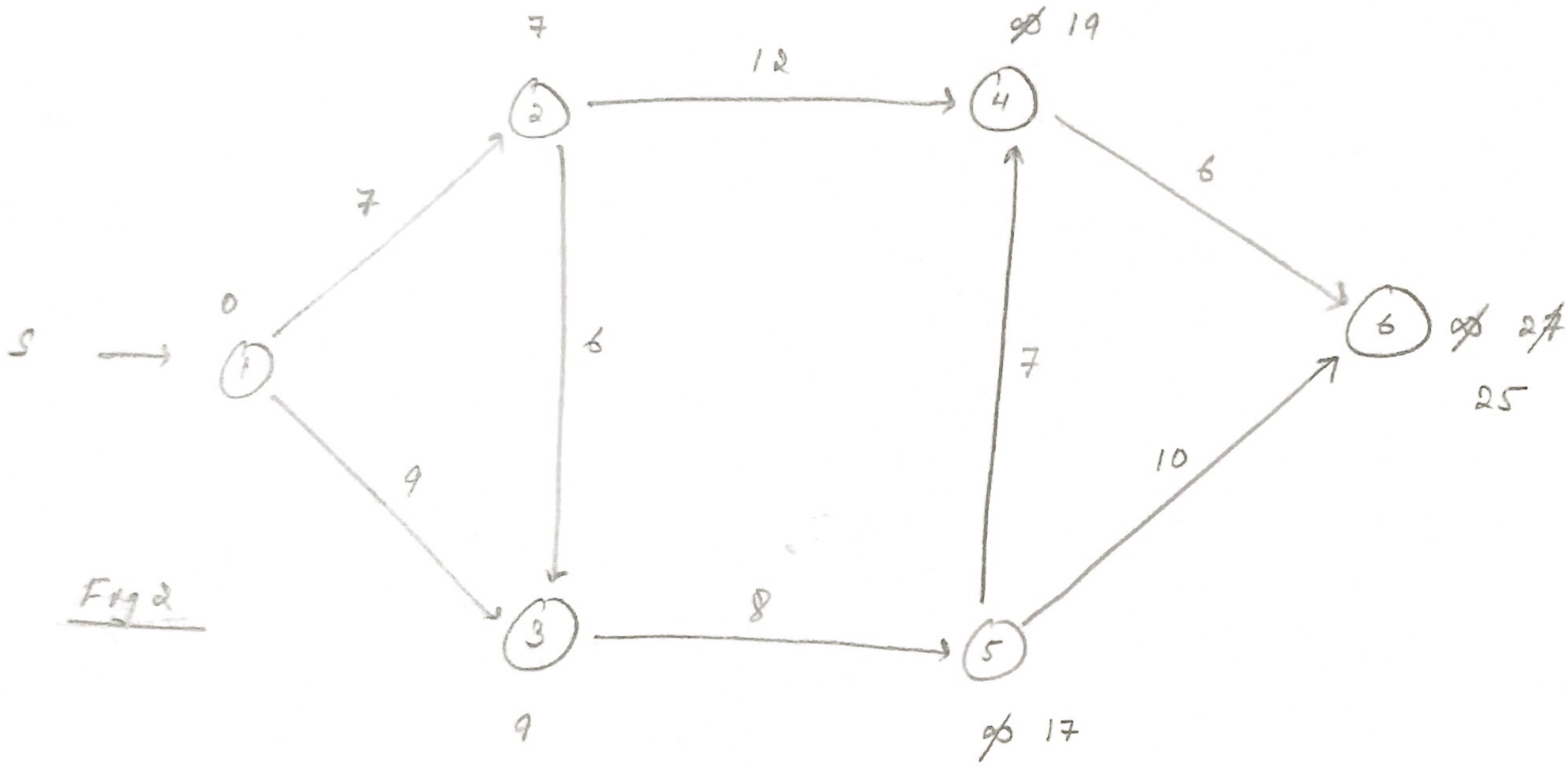
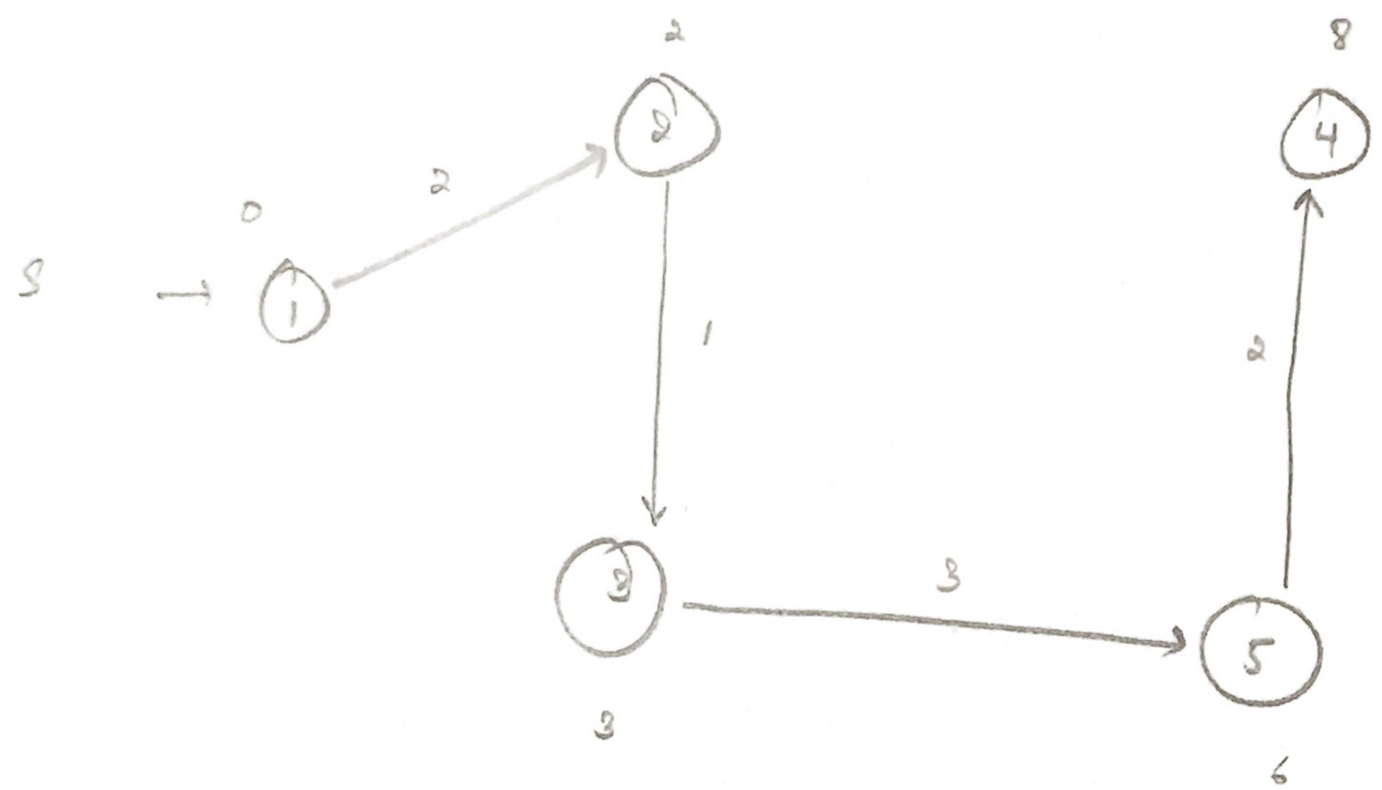


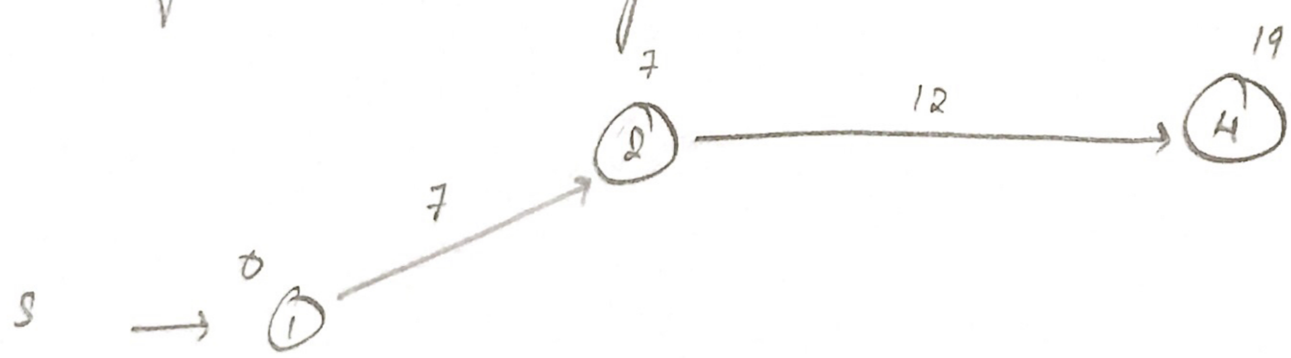
Fig 2

s	v	d[v]
1	2	7
1	3	9
1	4	19
1	5	17
1	6	25

Now if we wanna get from vertex (1) to vertex (4) with single source shortest path according to fig 1 then path looks like this:



Similarly, if we wanna get from vertex (1) to (4) with single source shortest path according to Fig 2 then path looks like this:



∴ It is proved that single-source shortest path tree from a source vertex (s) does not necessarily need to remain the same.