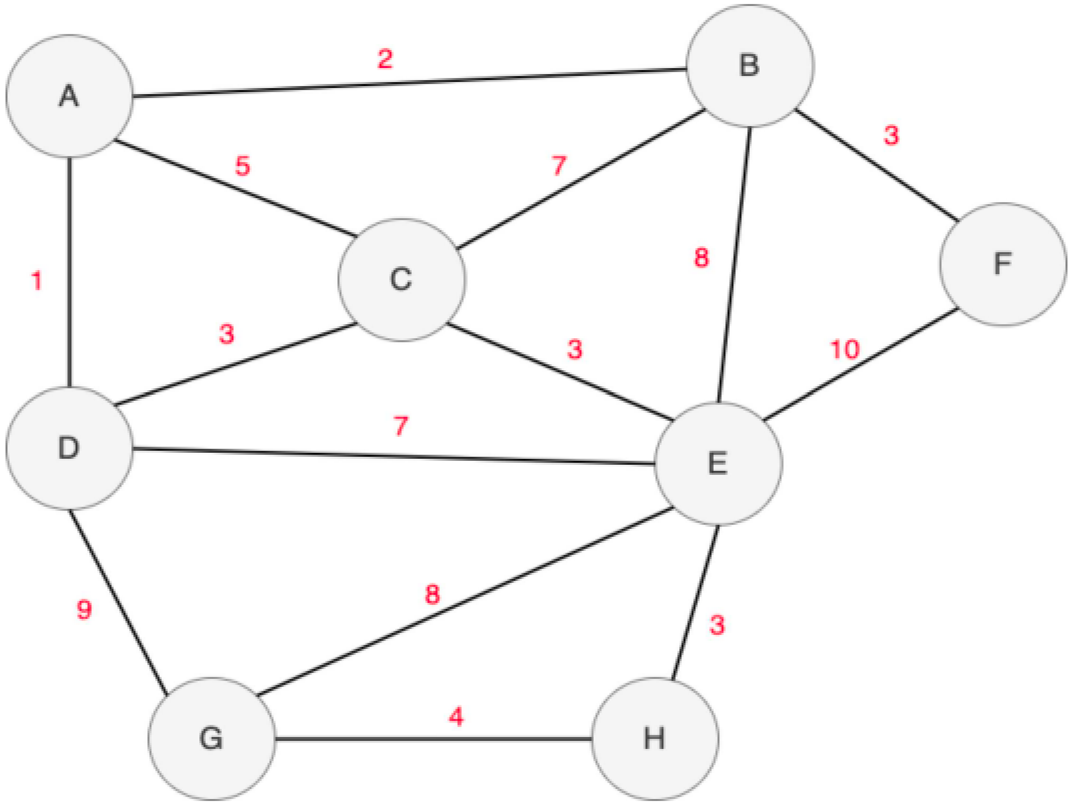
Yasin Ughur

28554

CS300 - Data Structures

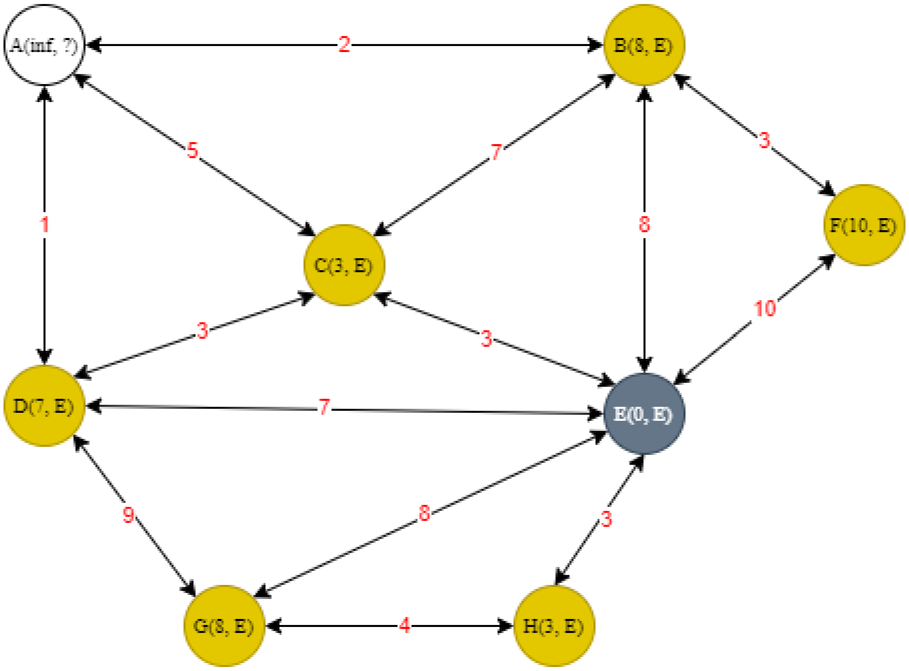
Homework 5

Question 1: Trace the operation of Dijkstra's weighted shortest path algorithm for the following graph. Use vertex E as your start vertex.

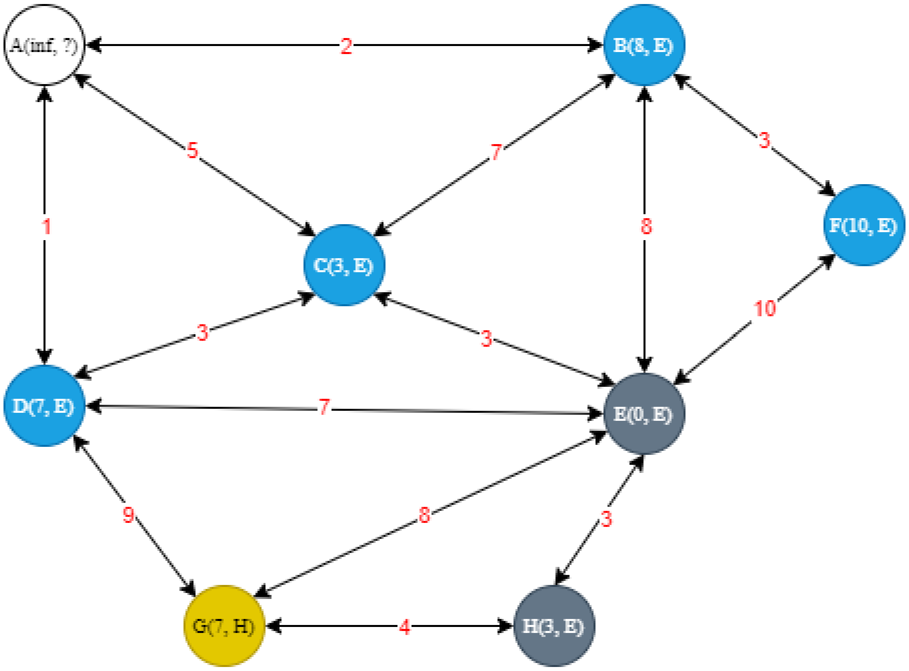


Solution:

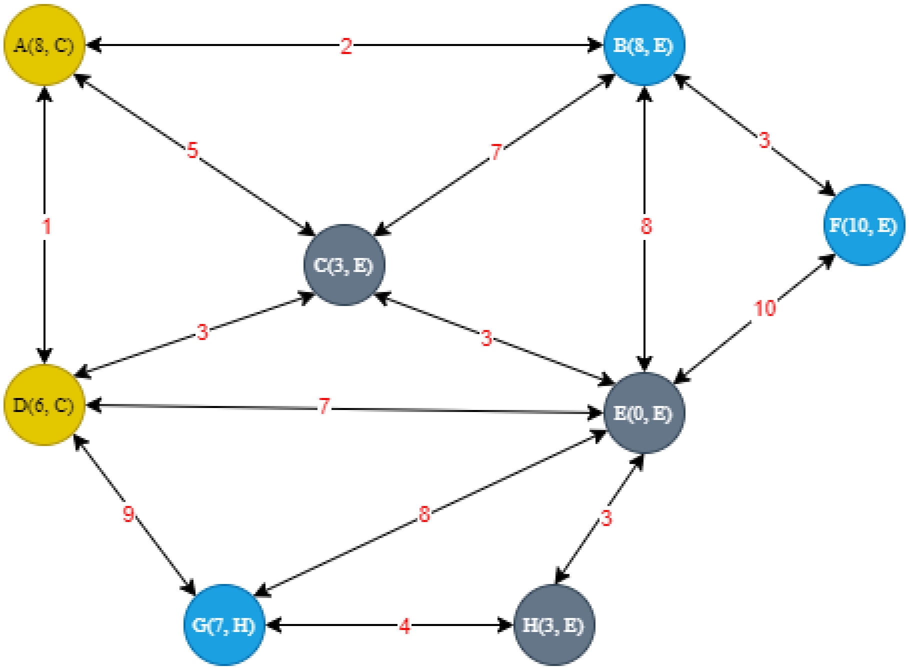
1. Select Vertex E for starting point. Update adjacent vertices.



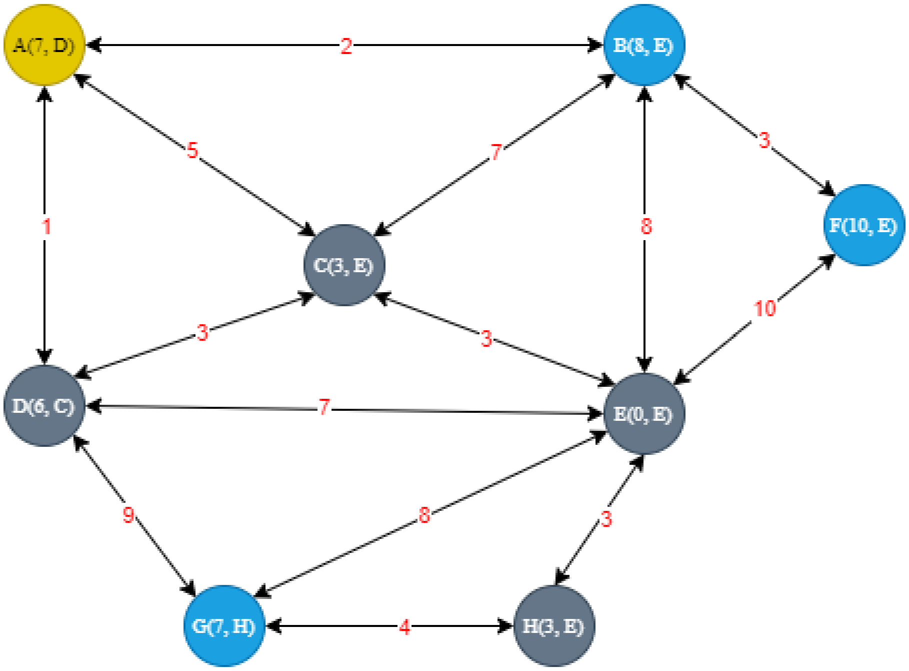
2. Select one with the smallest distance. For this case we can select C or H. I chose H and updated the adjacent vertices' distances.



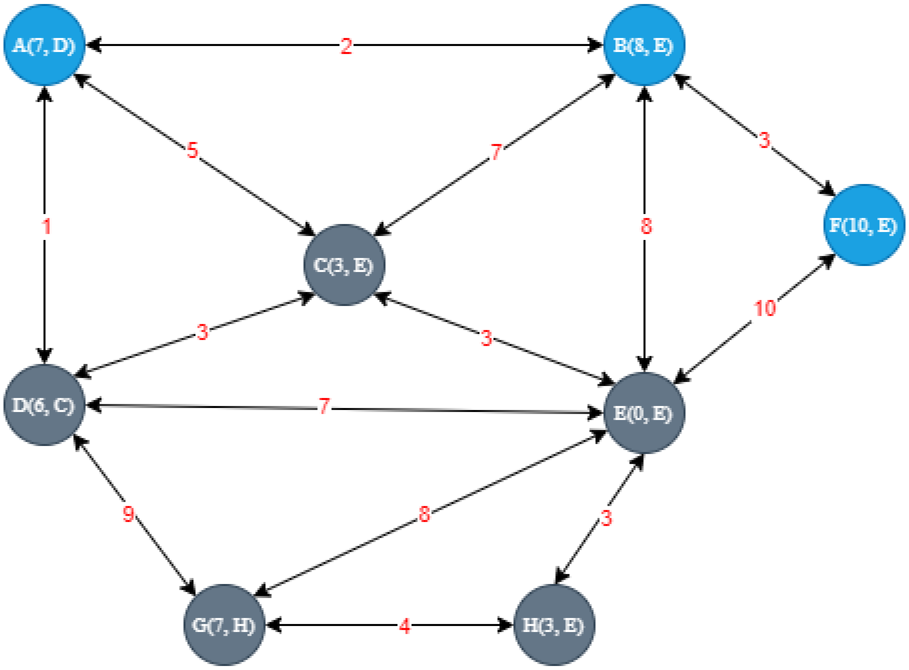
3. Select one with the shortest distance. For this case we can select C. I chose C and updated the adjacent vertices of C distances.



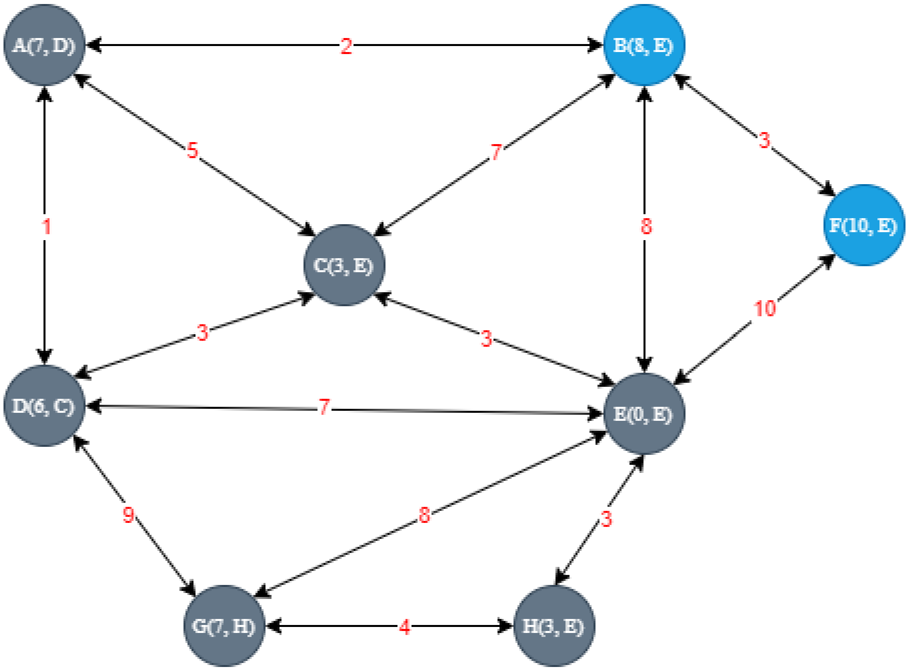
4. Select one with the shortest distance. For this case we can select D. There is not any need to update adjacent vertices.



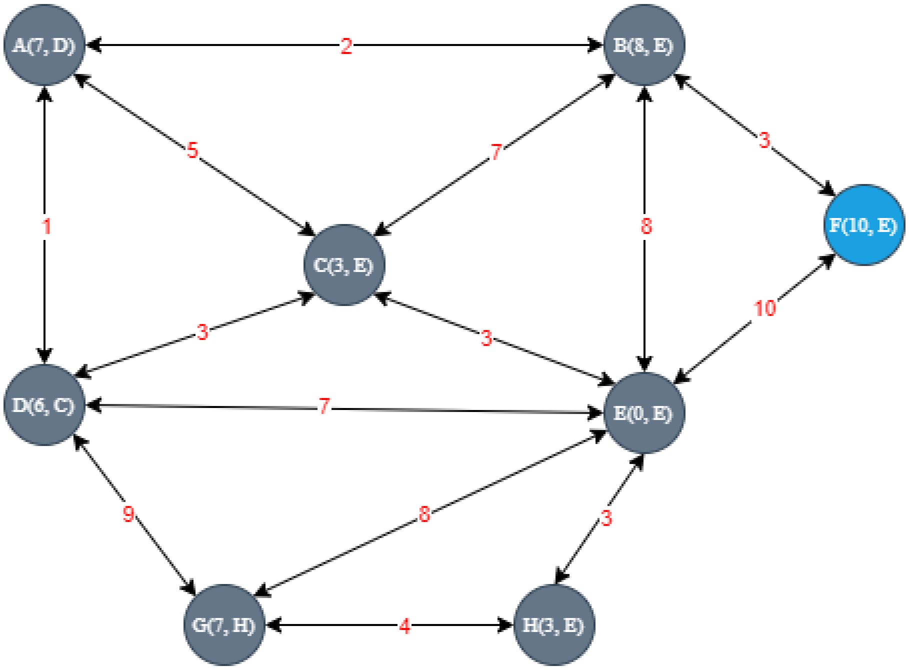
5. Select one with the shortest distance. For this case we can select D. There is not any need to update adjacent vertices.



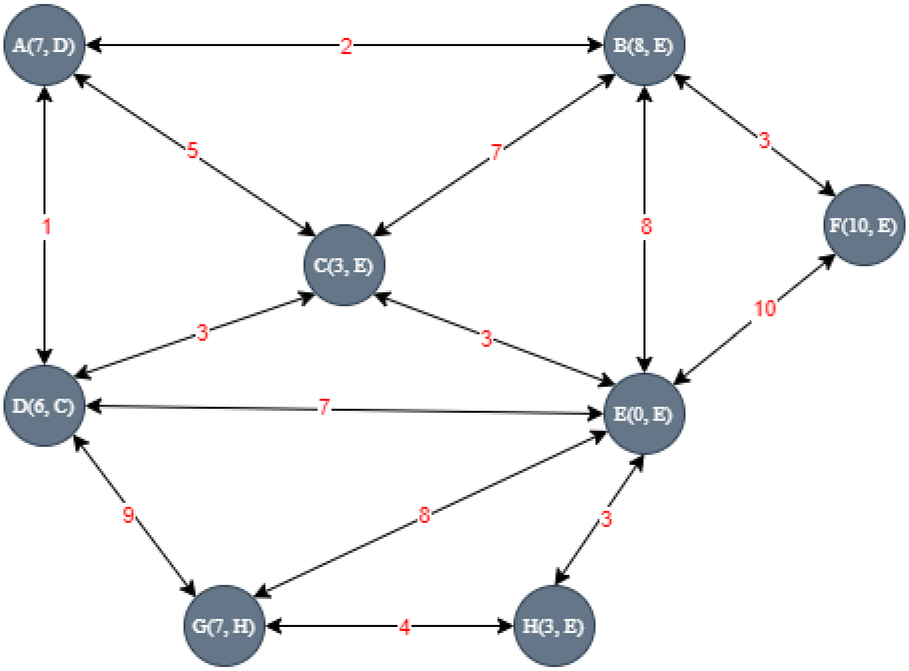
6. Select one with the smallest distance. For this case we can select A or B. I chose A and updated the adjacent vertices' distances.



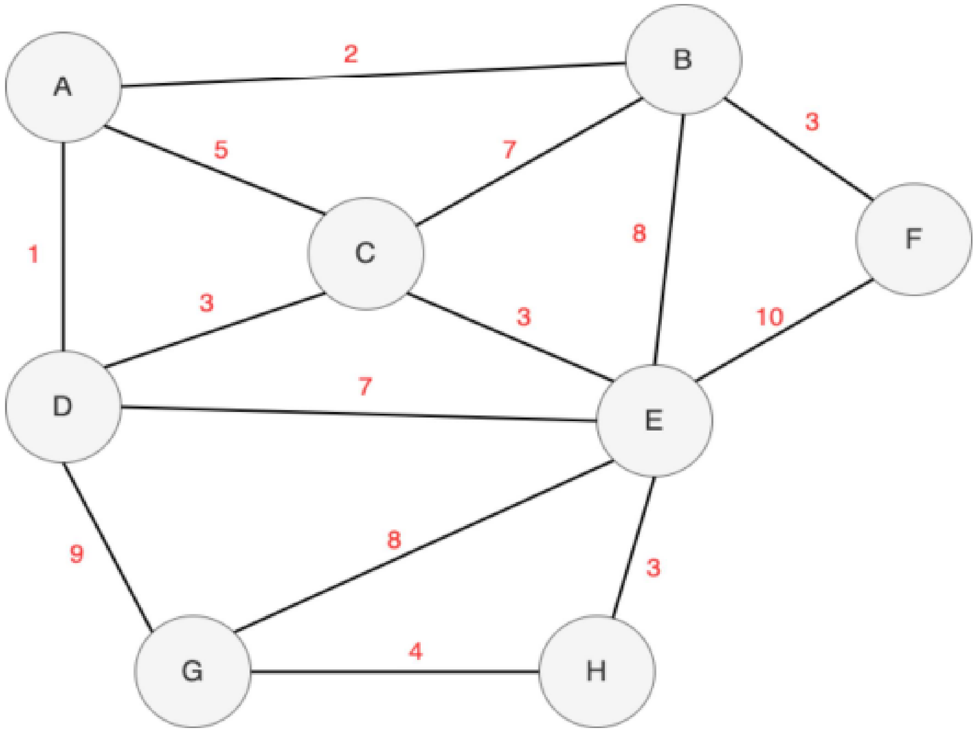
7. Select one with the smallest distance. For this case we can select A. I chose A and updated the adjacent vertices' distances.



8. Select one with the smallest distance. For this case we can select F. There are not any other unknown vertices left. The program is over.

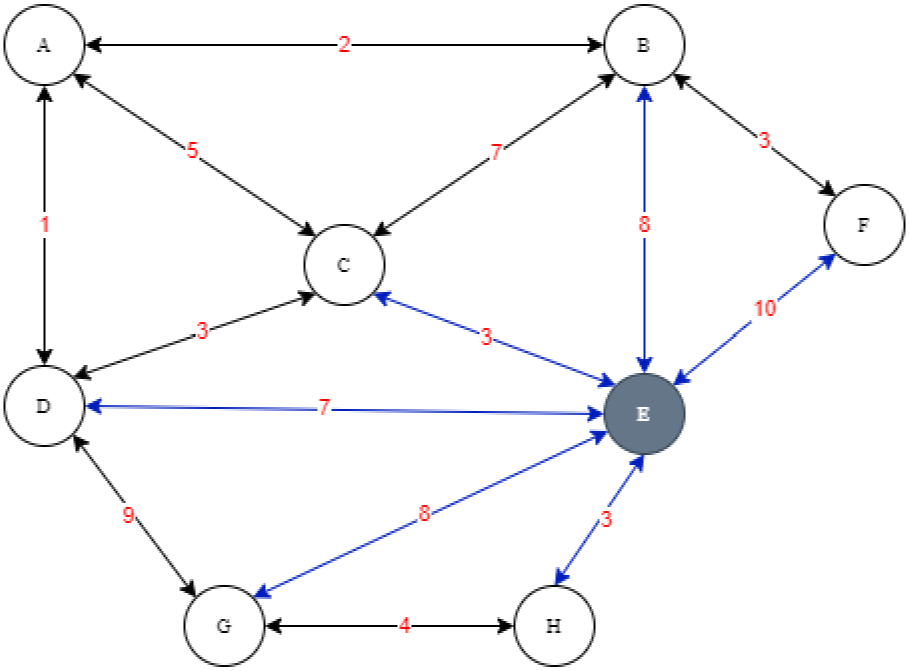


Question 2: Trace the operation of Prim's minimum spanning tree algorithm for the graph in Figure 1. Use vertex E as your start vertex.

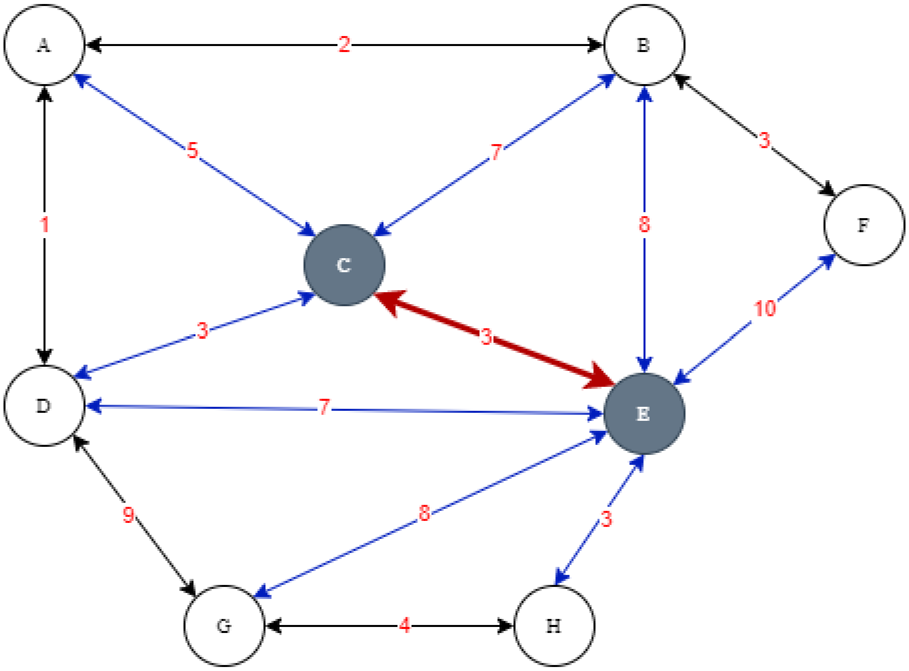


Solution:

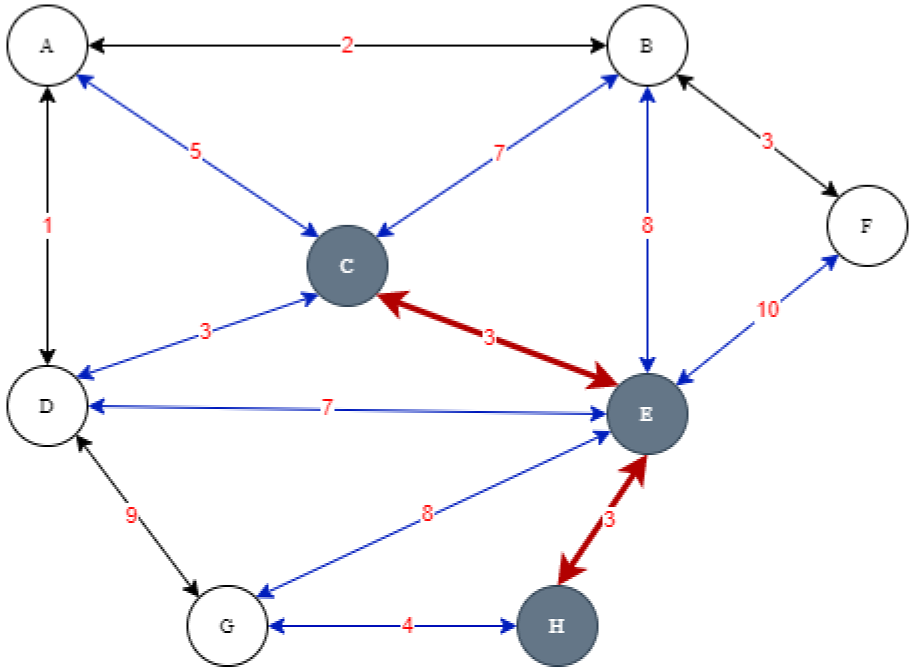
1. We can start from vertex E and make it as known. After that, pick the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, pick E-C and mark C as known.



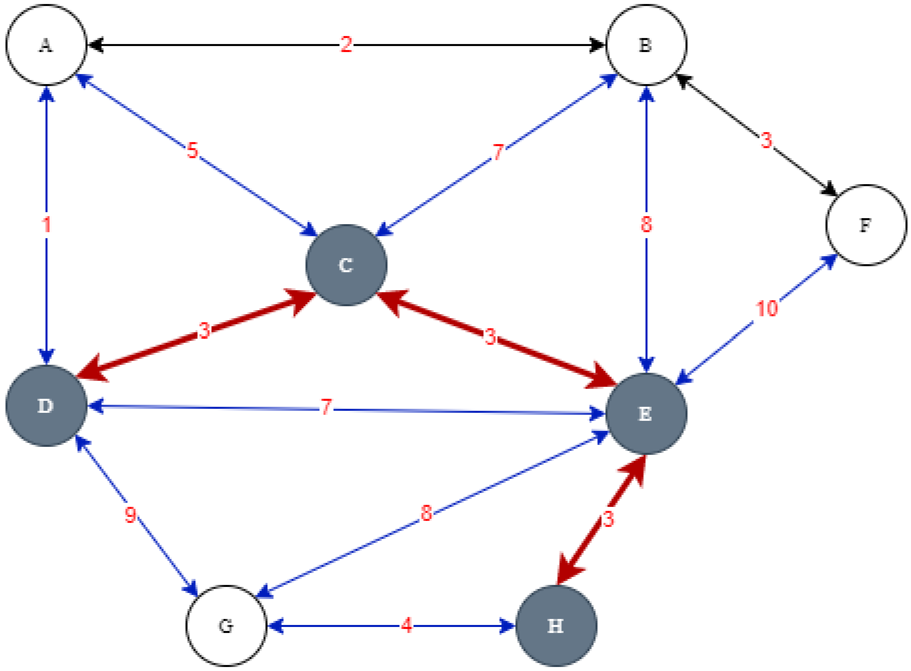
2. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, E-H and mark H as known.



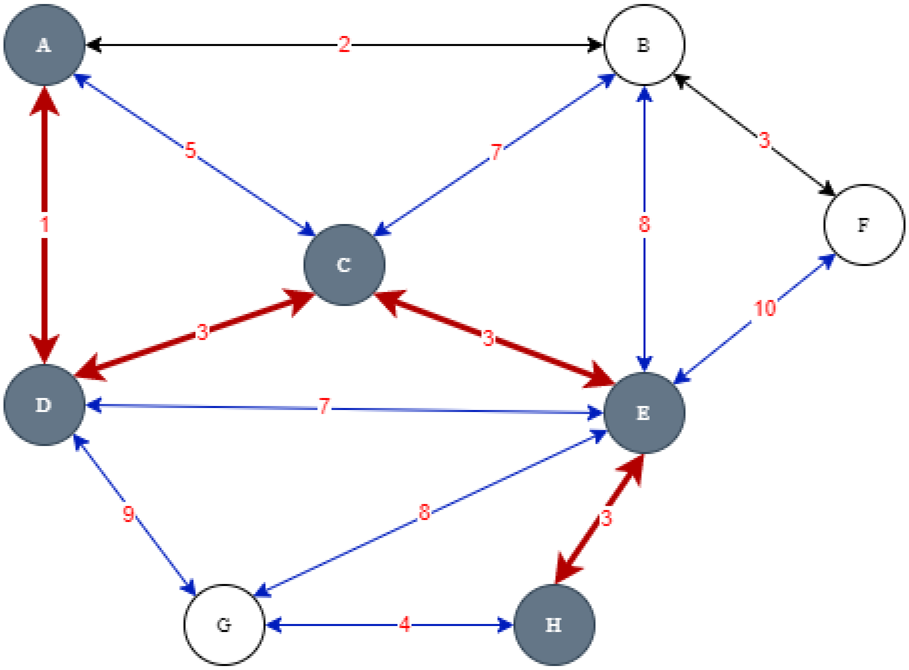
3. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, C-D and mark D as known.



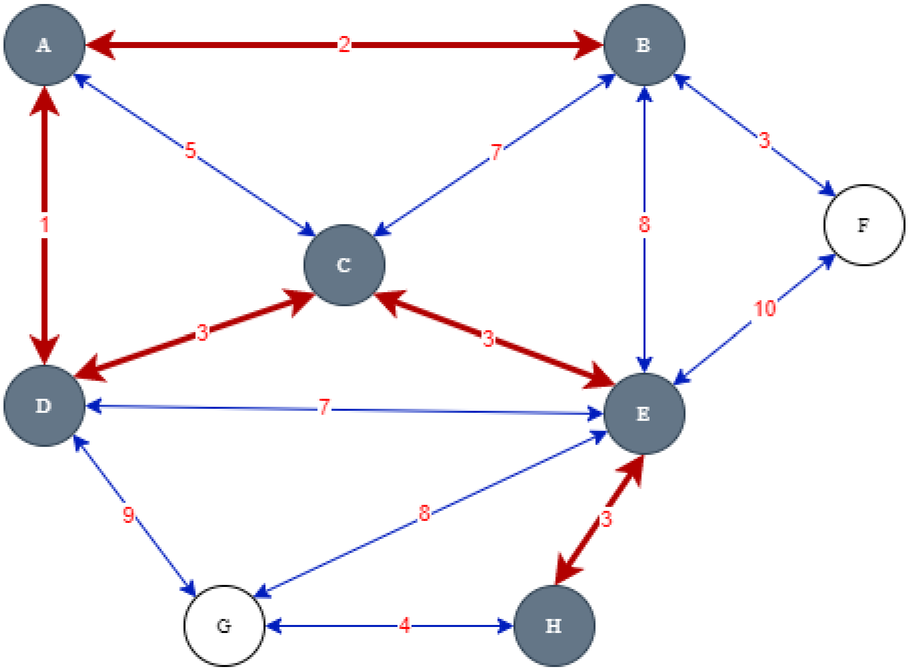
4. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, D-A and mark A as known.



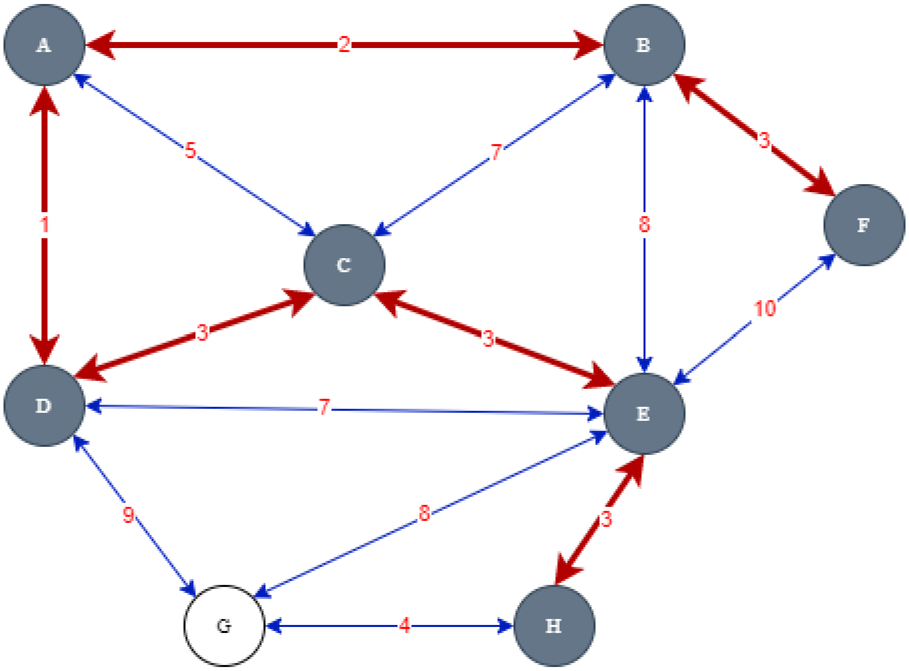
5. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, A-B and mark B as known.



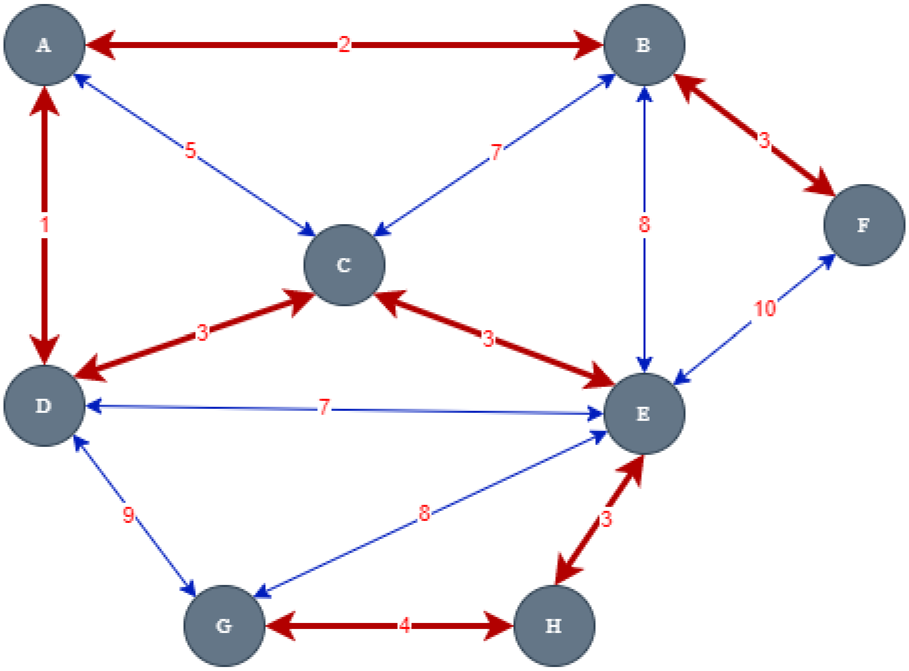
6. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, B-F and mark F as known.



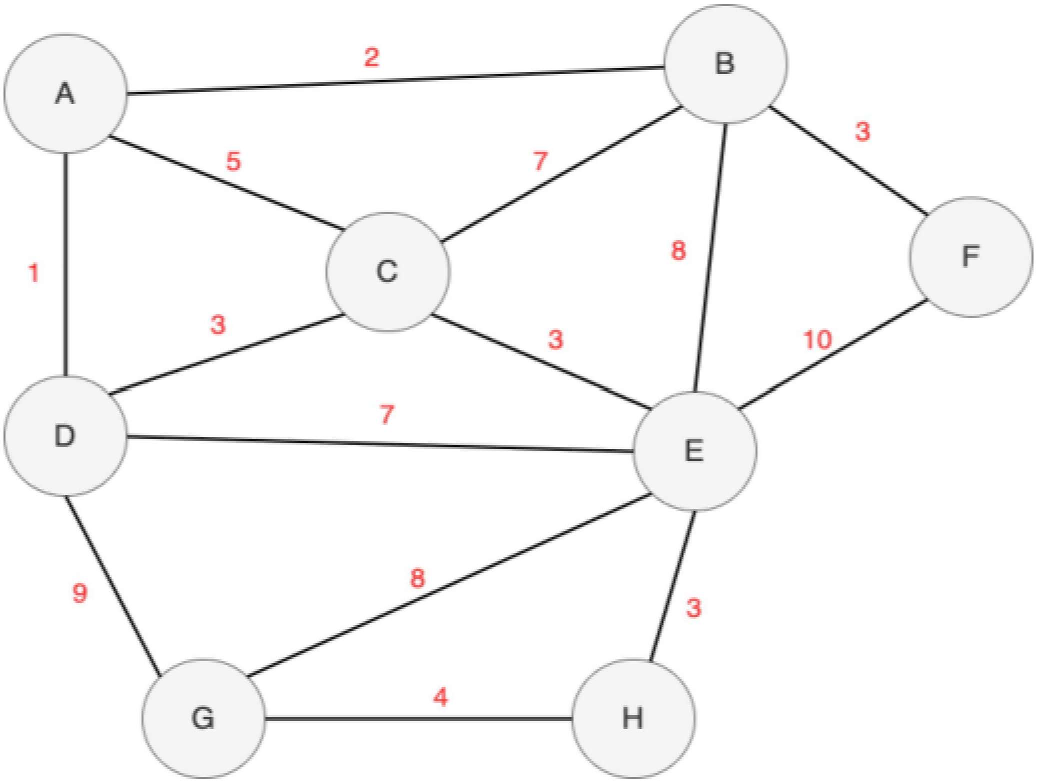
7. Get the edge with the lowest weight which connects with known vertex and unknown vertex. For this case, H-G and mark G as known.



8. As you can see, all vertices are included in the spanning tree. The program is over.

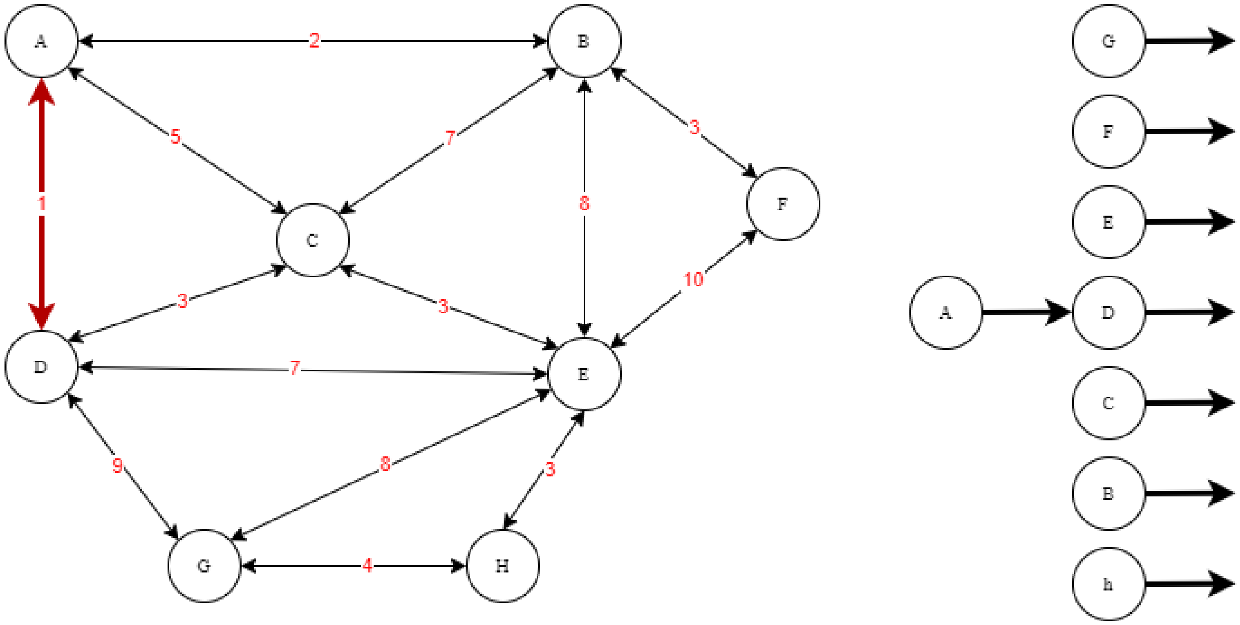


Question 3: Trace the operation of Kruskal's minimum spanning tree algorithm for the graph in Figure 1.

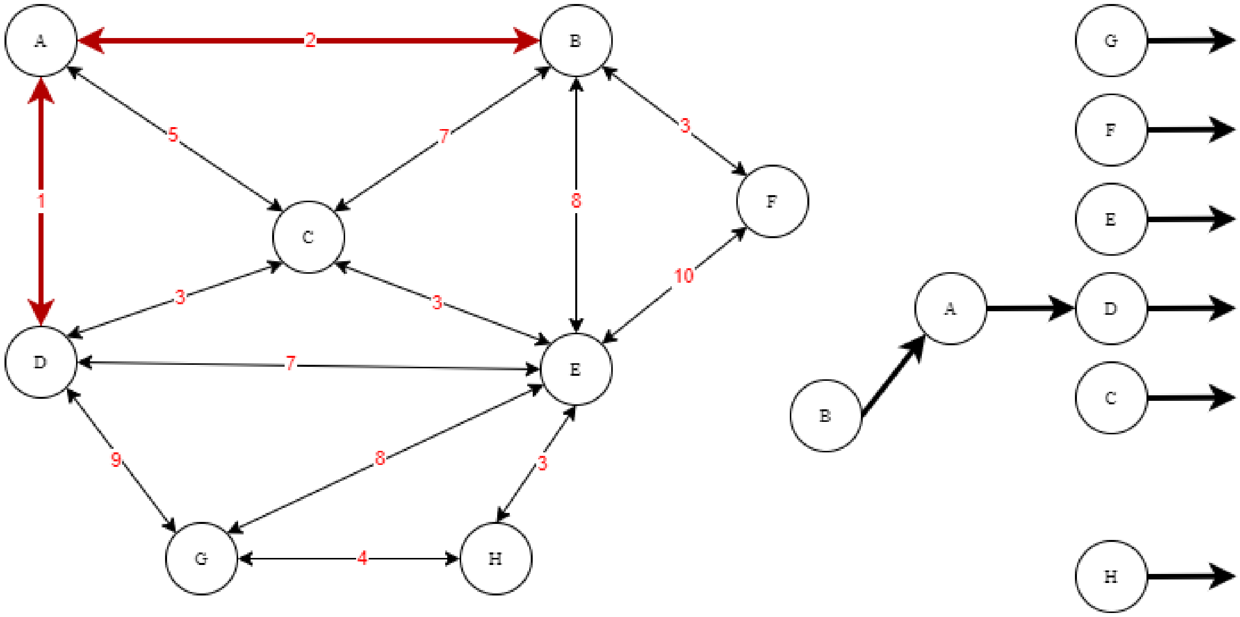


Solution:

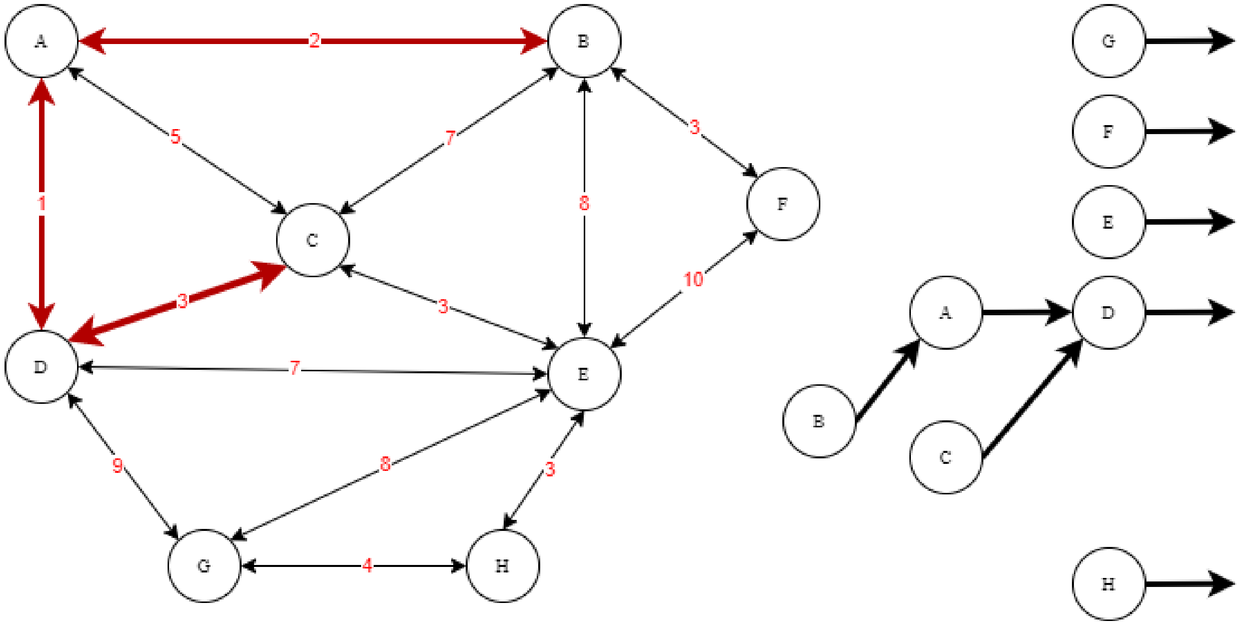
1. Get the edge with the smallest weight. In this case, this is A-D with weight 1. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.



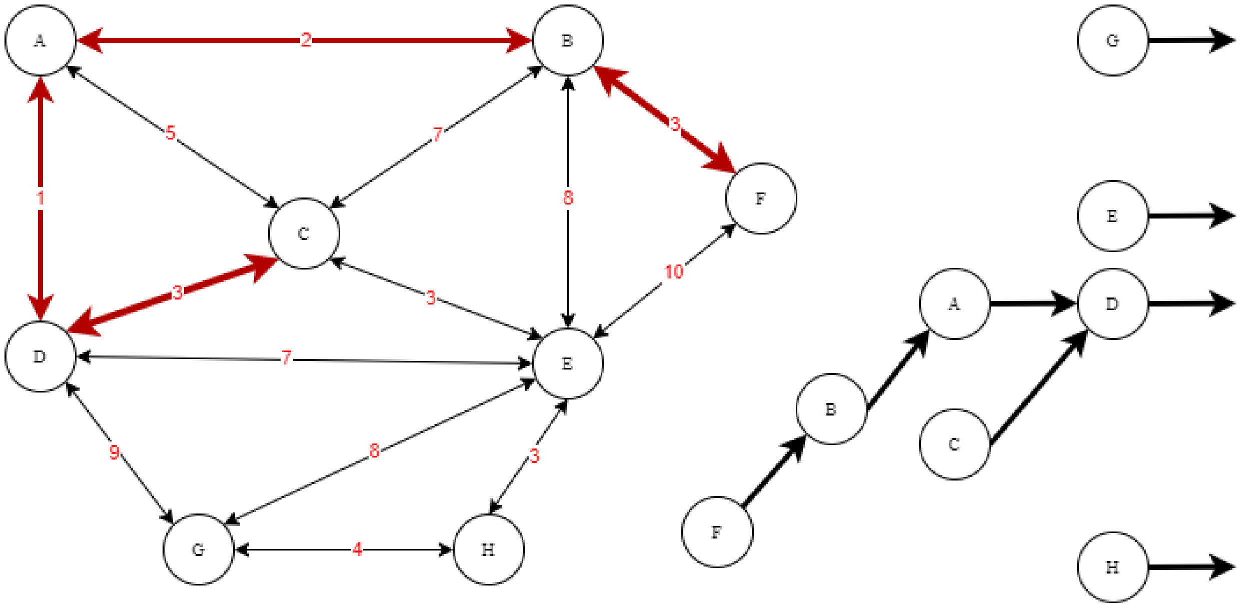
2. Get the edge with the smallest weight. In this case, this is A-B with weight 2. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.



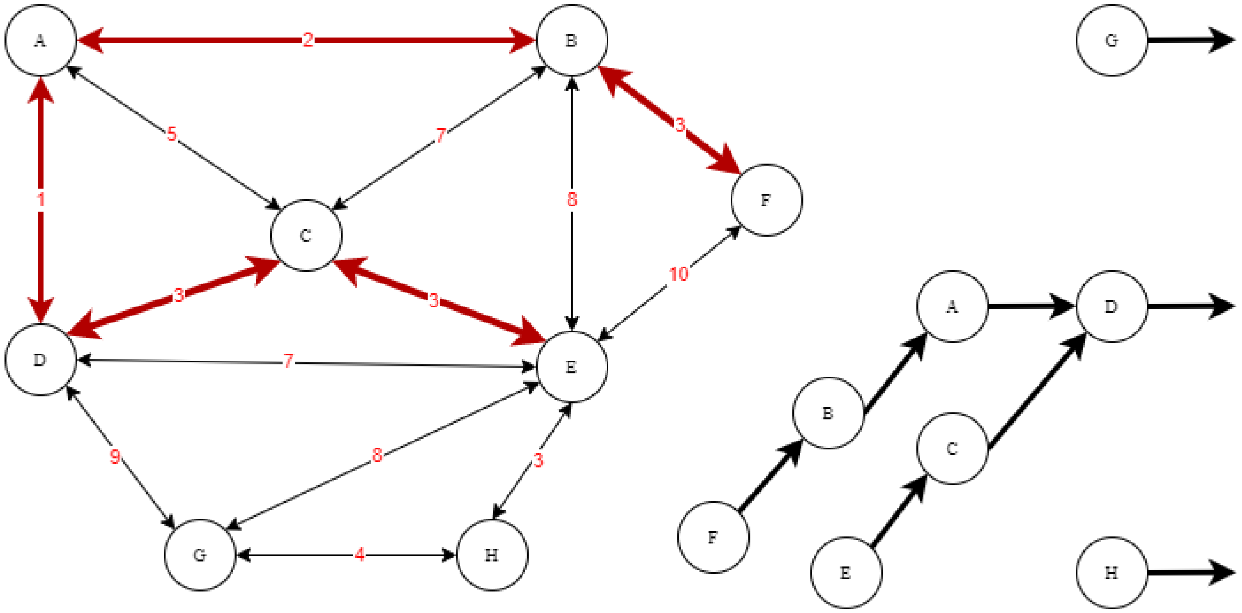
3. Get the edge with the smallest weight. In this case, these are D-C, E-C, E-H, B-F with weight 3. Firstly, select D-C. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.



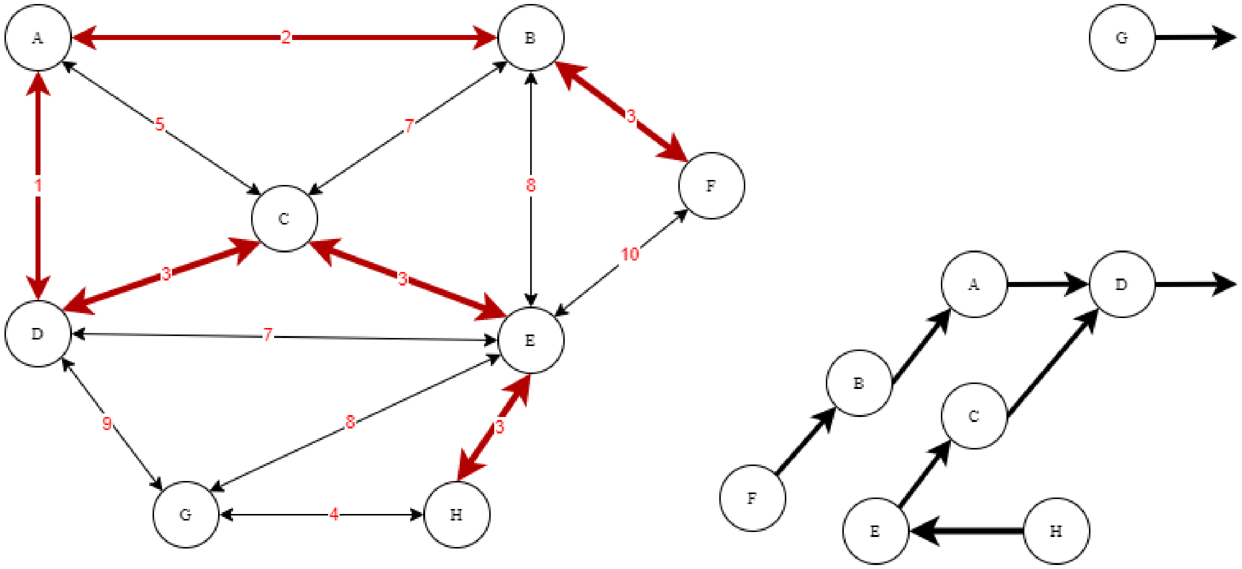
4. Get B-F which weighs 3. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.



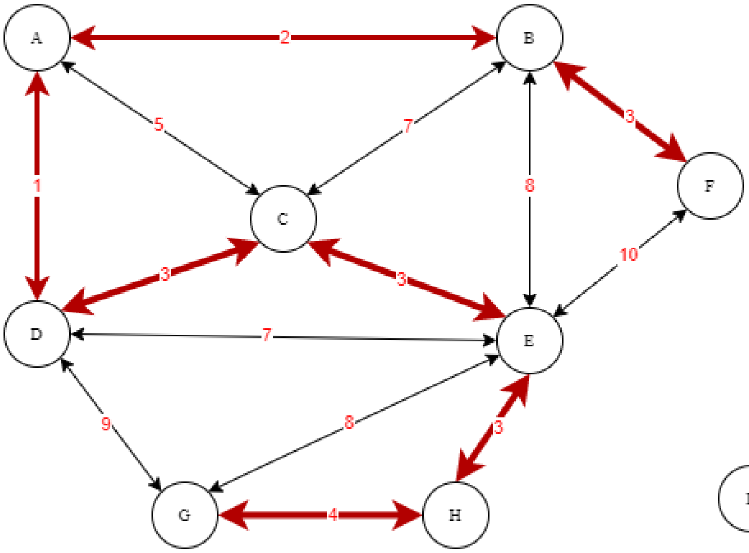
5. Get C-E which weighs 3. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.

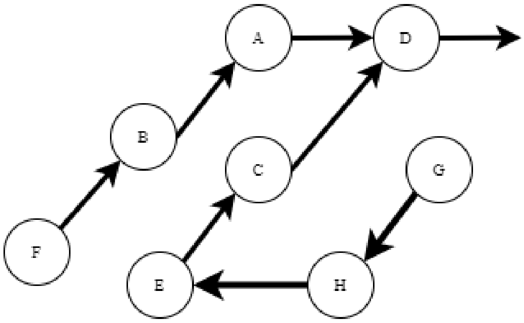


6. Get E-H which weighs 3. With the help of union operation we can union them. Then add the edge to the minimum spanning tree.

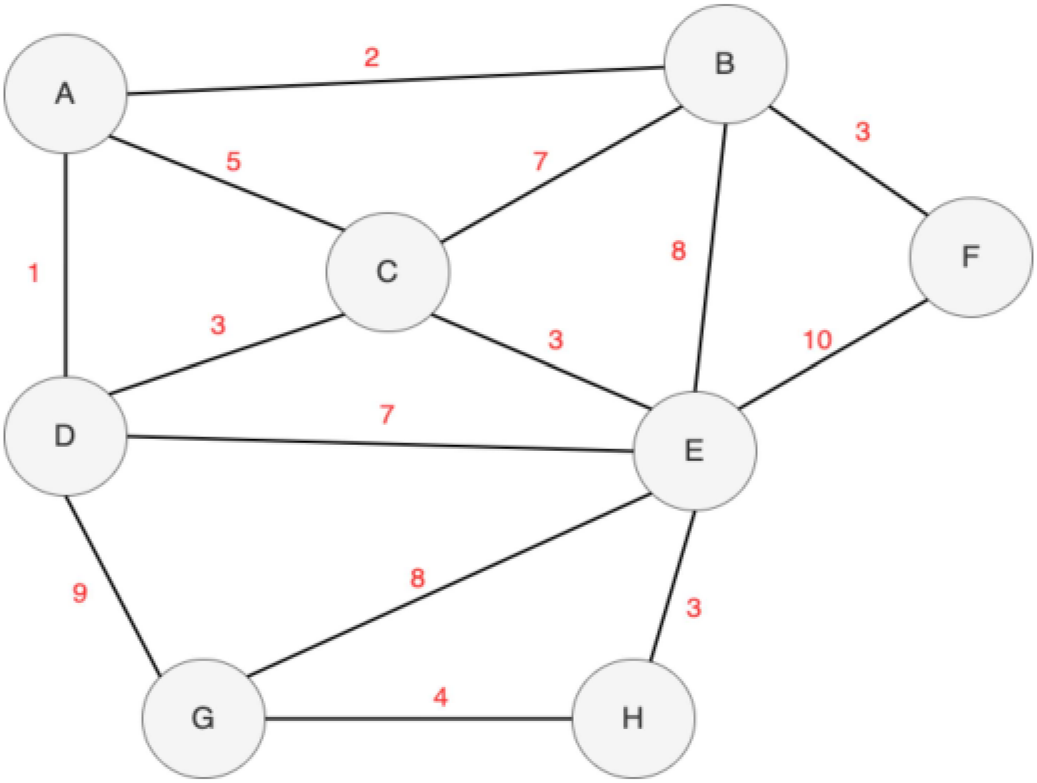


7. Get the edge with the smallest weight. In this case, this is H-G with weight 4. With the help of union operation we can union them. Then add the edge to the minimum spanning tree. Now we have only one tree. The minimum spanning tree covers all the vertices. The program is over.


A
B
C
D
E
F
G
H
2
5
7
1
8
3
3
3
10
7
9
8
3
4

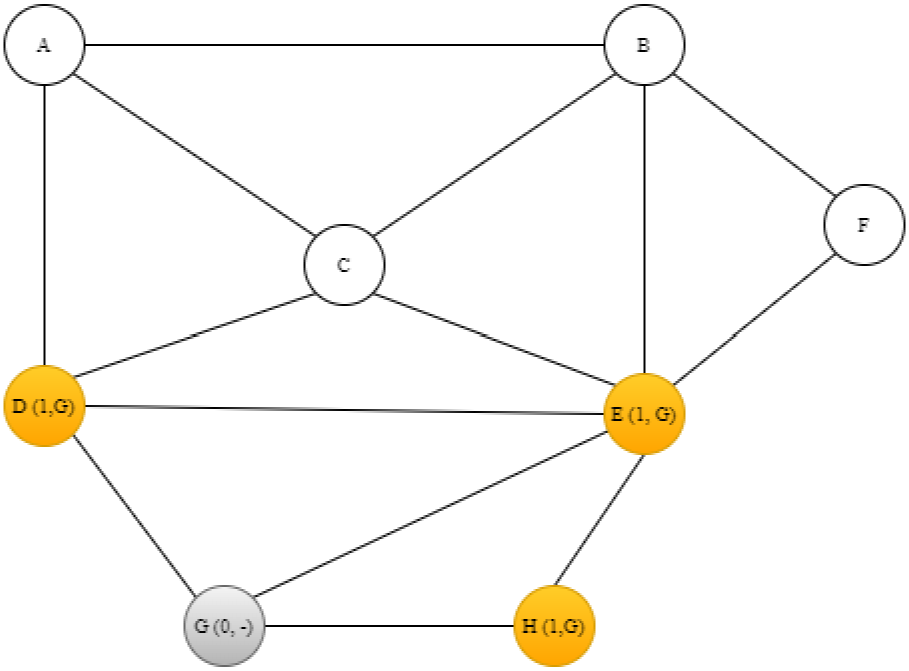

A
D
B
C
G
F
E
H

Question 4: Find shortest unweighted path from G to all other vertices for the graph in Figure 1. Use breadth-first search algorithm in your answer. Do NOT forget to show the trace.

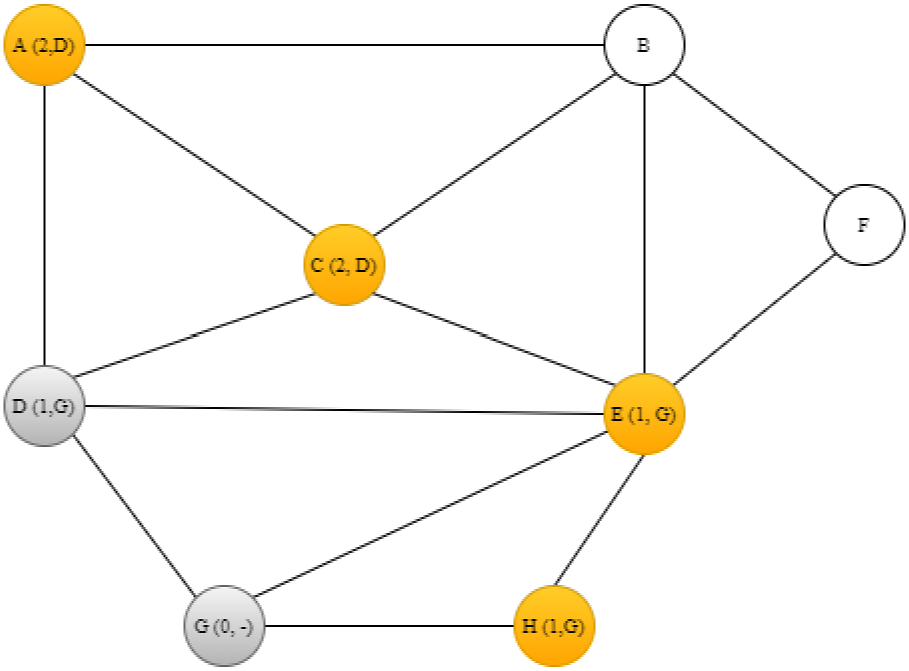


Solution:

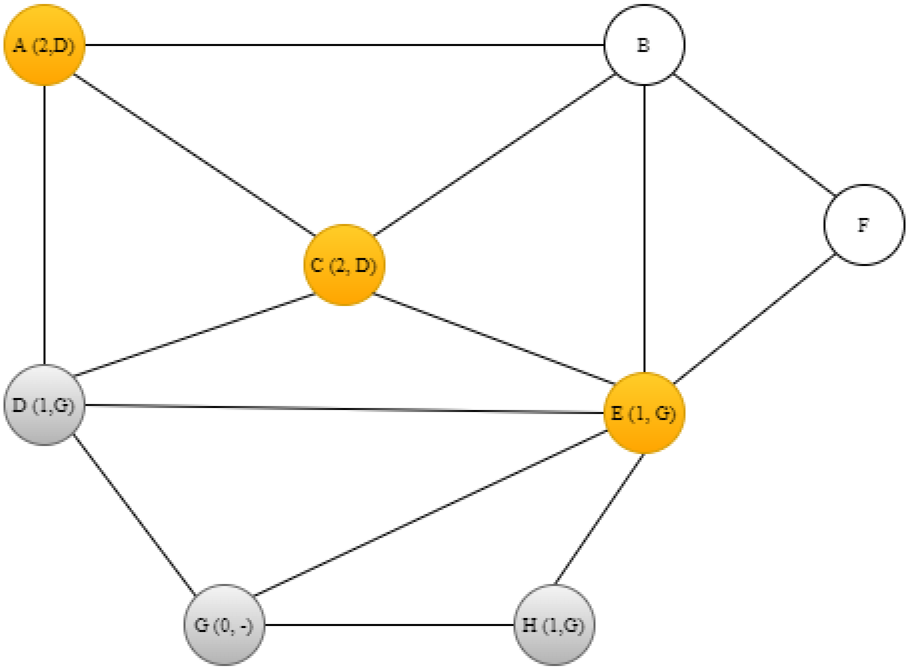
1. Firstly, make G as visited and then update all unknown vertices adjacent to its distances as 1.



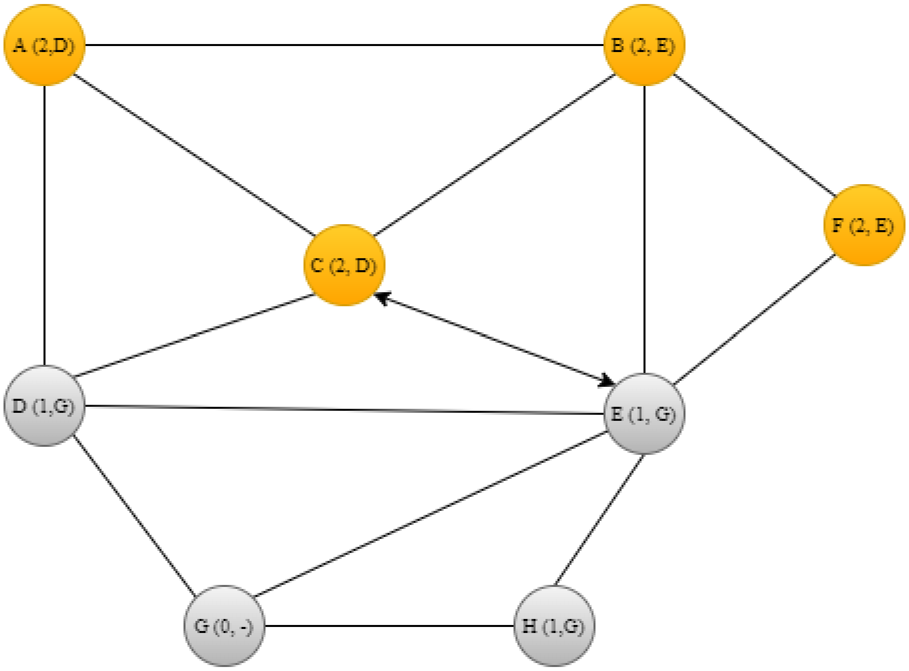
2. Secondly, make G as visited and then update all unknown vertices adjacent to its distances as 2.



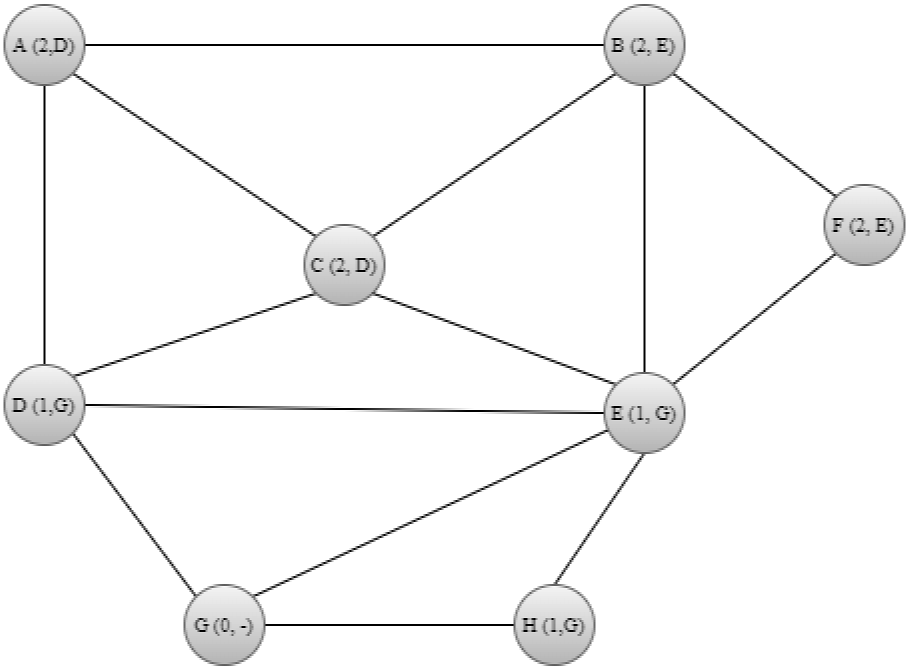
3. Thirdly, make H as visited.



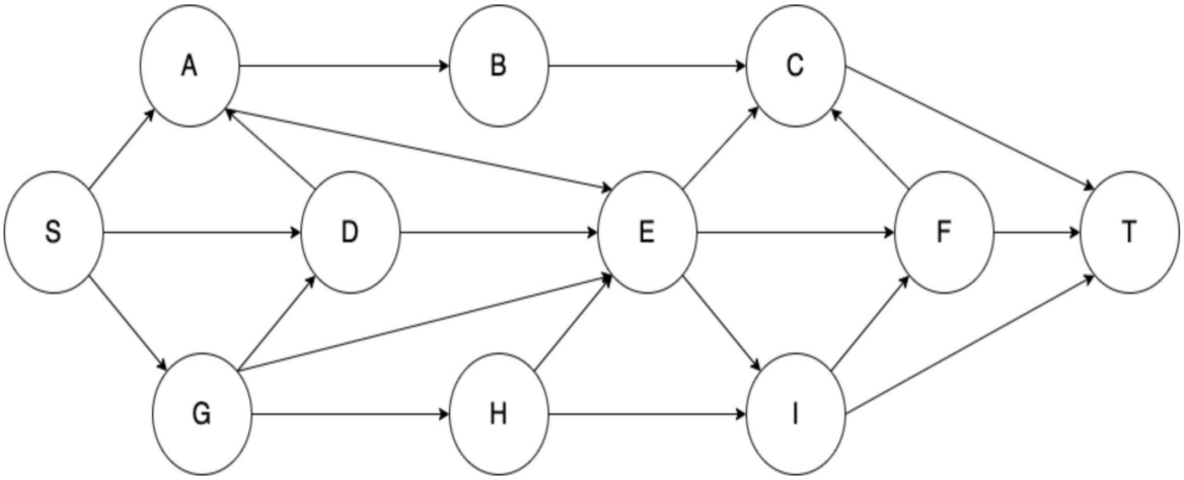
4. Fourthly, make E as visited and then update all unknown vertices adjacent to its distances as 2.



5. As you can see, there are no unknown vertices. For this reason, make all vertices as visited.

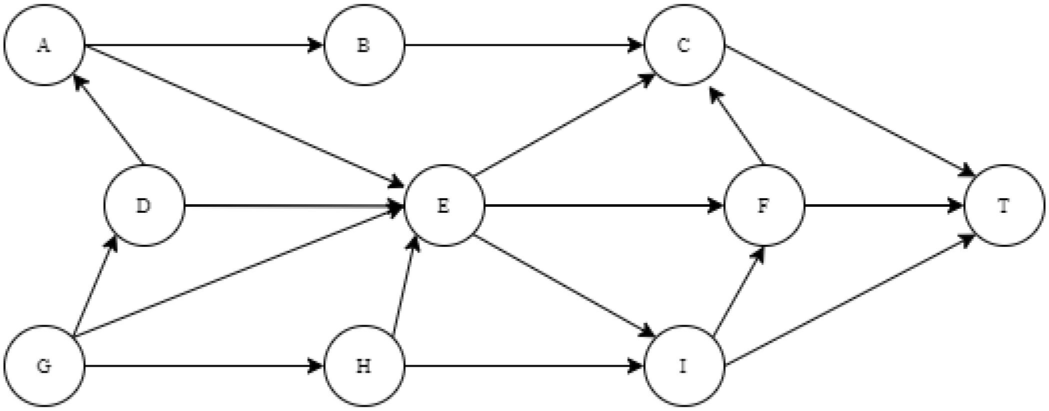

A (2,D)
B (2, E)
F (2, E)
C (2, D)
D (1,G)
E (1, G)
G (0, -)
H (1,G)

Question 5: Find a topological ordering of the graph in Figure 2.



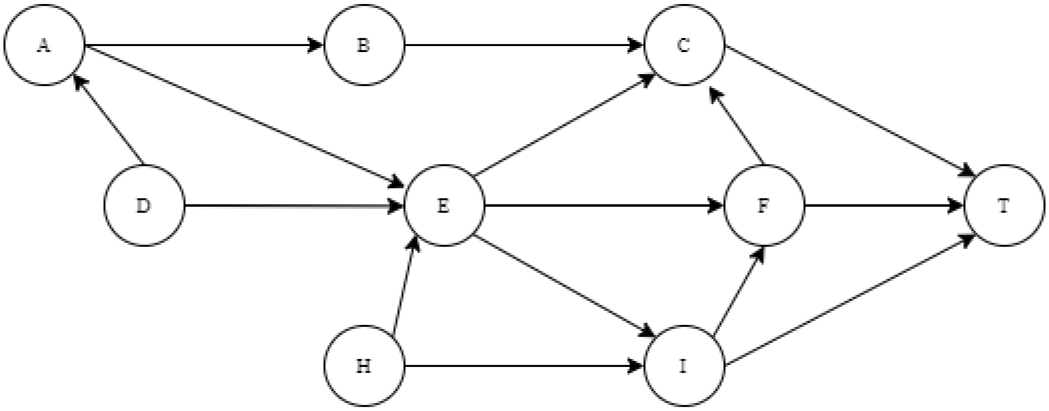
Solution:

1. Get a vertex with indegree 0 (S). Firstly, print then remove it.



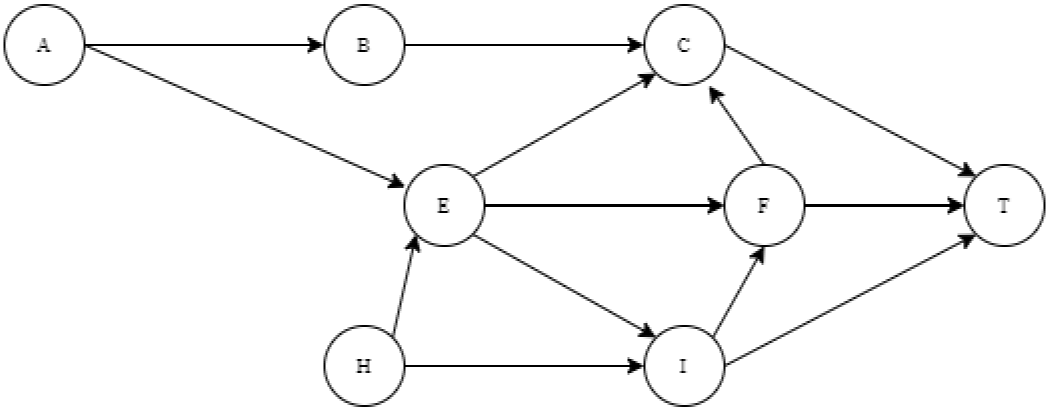
S-

2. Get a vertex with indegree 0 (G). Firstly, print then remove it.



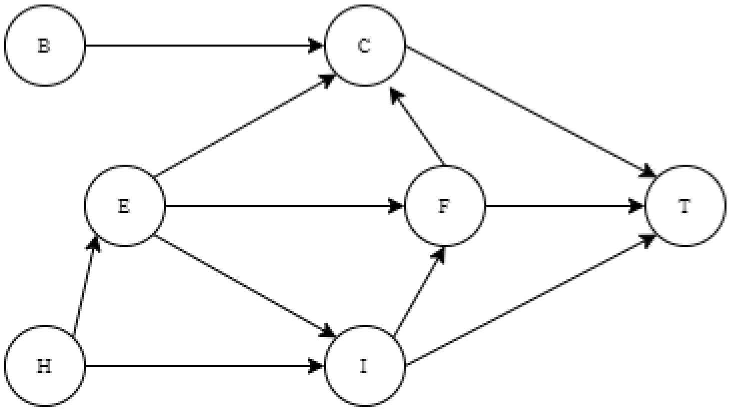
S-G-

3. Get (D). Firstly, print then remove it.



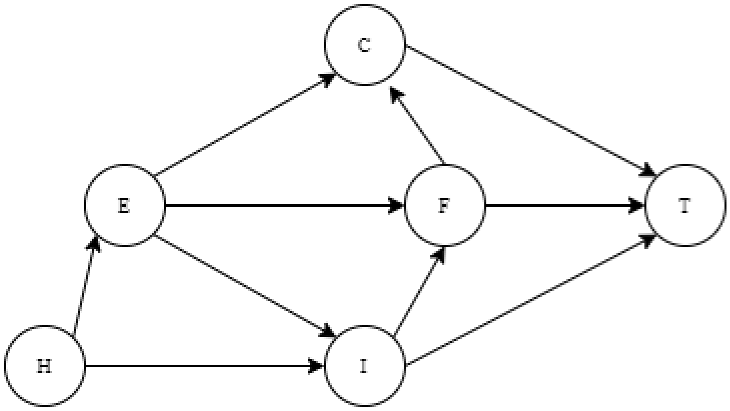
S-G-D-

4. Get (A). Firstly, print then remove it.



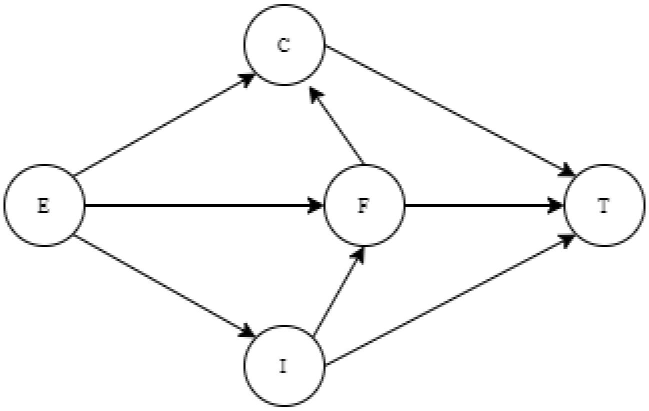
S-G-D-A

5. Get (B). Firstly, print then remove it.



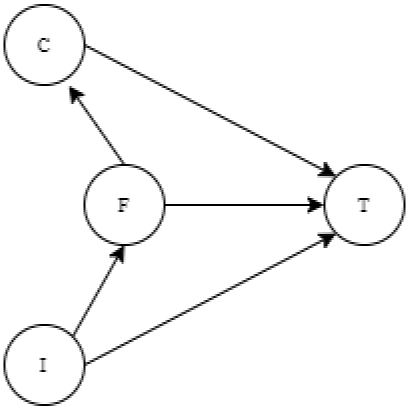
S-G-D-A-B

6. Get (H). Firstly, print then remove it.



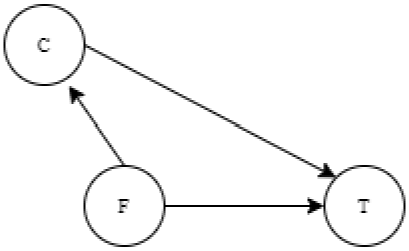
S-G-D-A-B-H

7. Get (E). Firstly, print then remove it.



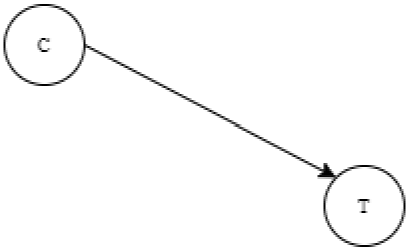
S-G-D-A-B-H-E

8. Get (I). Firstly, print then remove it.



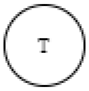
S-G-D-A-B-H-E-I

9. Get (F). Firstly, print then remove it.



S-G-D-A-B-H-E-I-F

10. Get (C). Firstly, print then remove it.



S-G-D-A-B-H-E-I-F-C

11. Get (T). Firstly, print then remove it. The program is over.
S-G-D-A-B-H-E-I-F-C-T