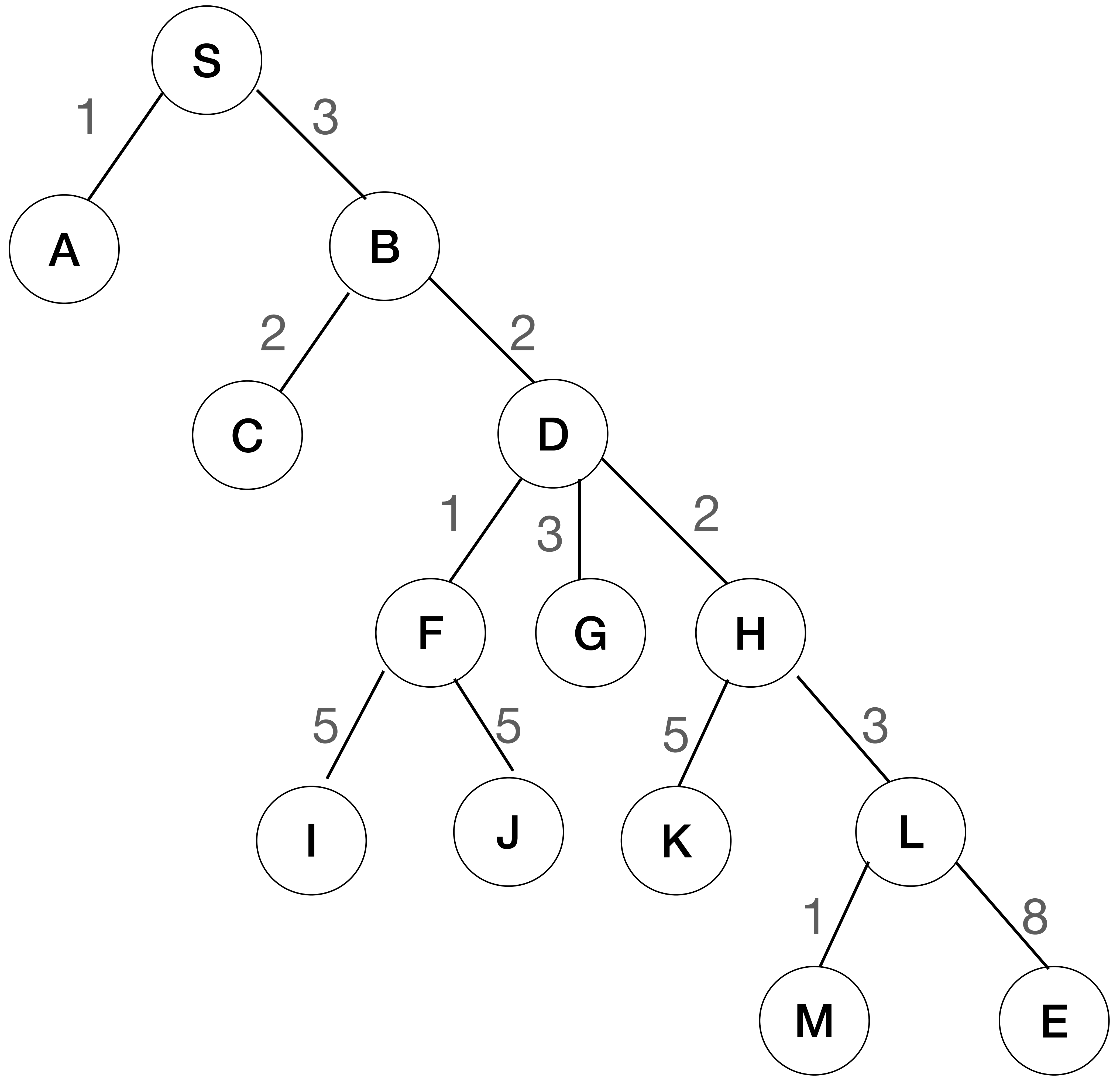
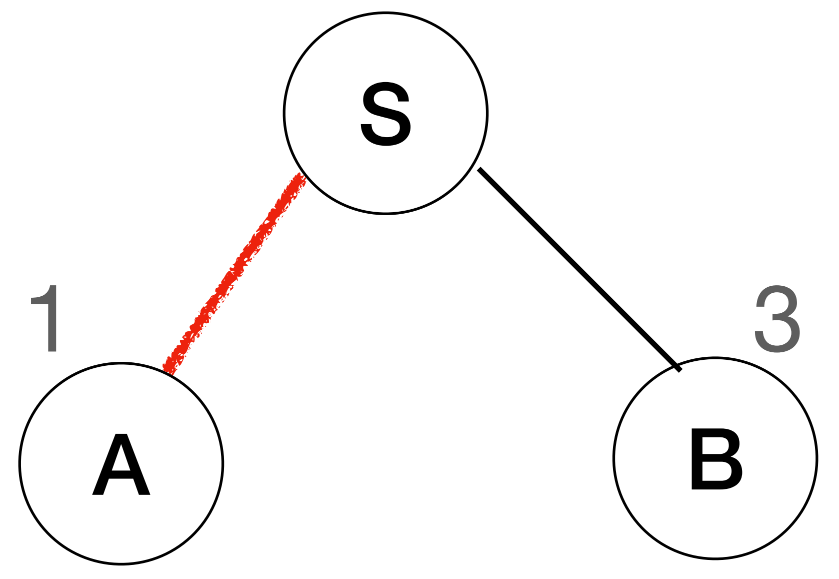
Q25: Use Prim's Minimum Spanning Tree algorithm and Kruskal's Minimum Spanning Tree algorithm to find the shortest path of a maze.

Tree representation of the maze:



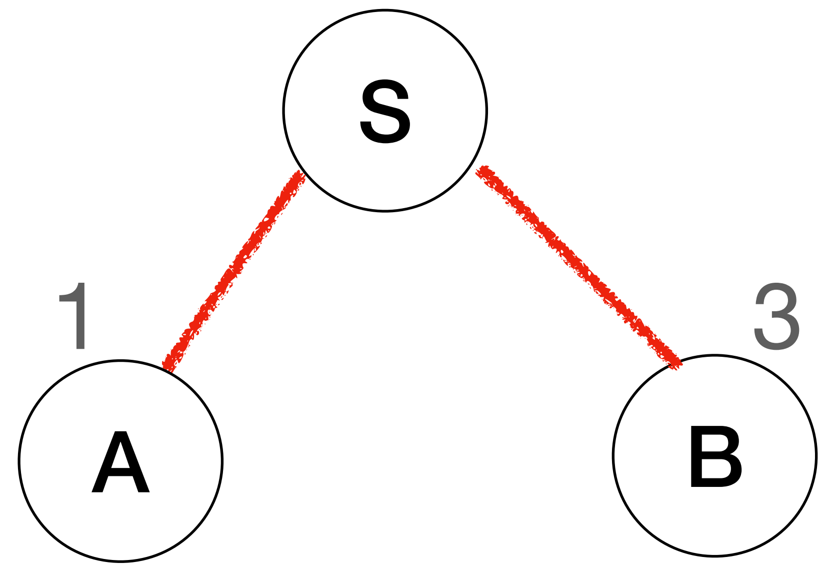
Step 1: Prim’s Minimum Spanning Tree algorithm

1.mstSet = {S}



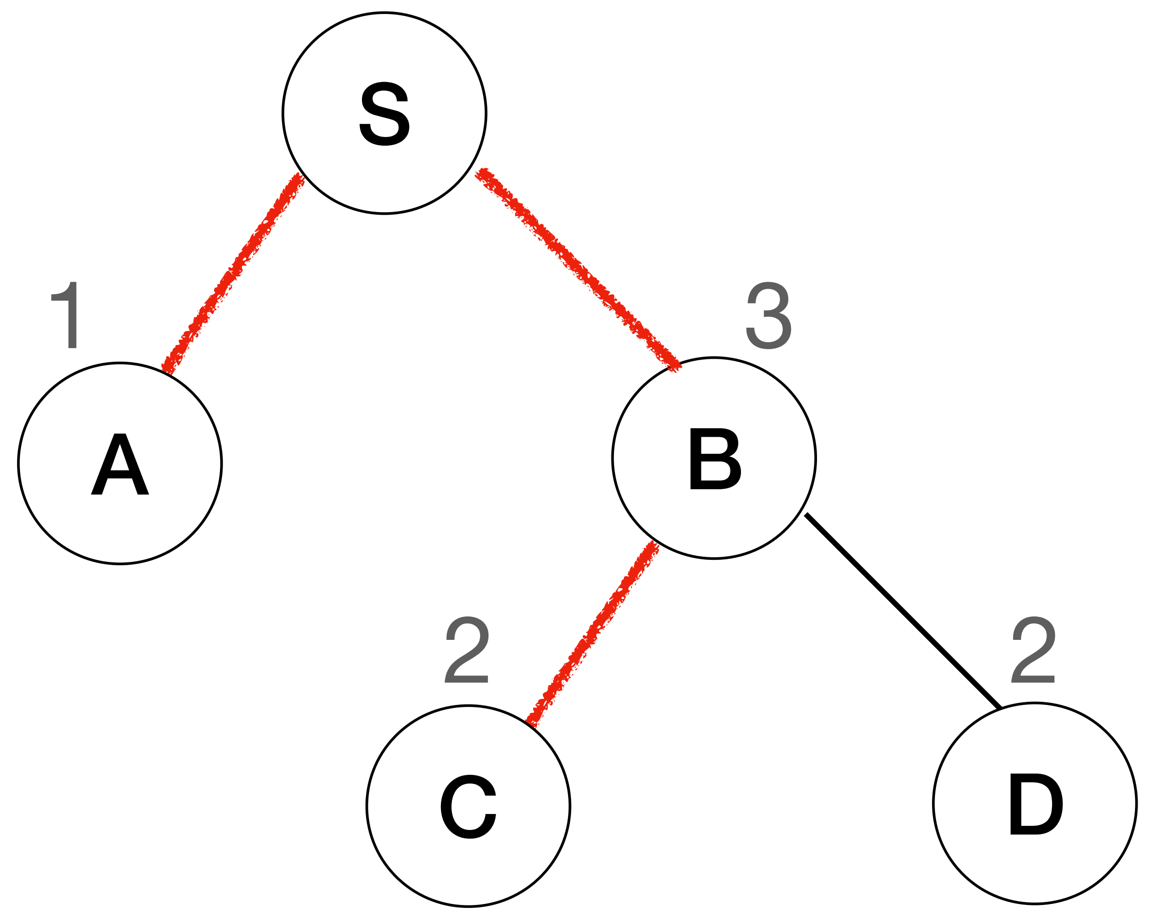
Add vertex A to mstSet

2.mstSet = {S,A}



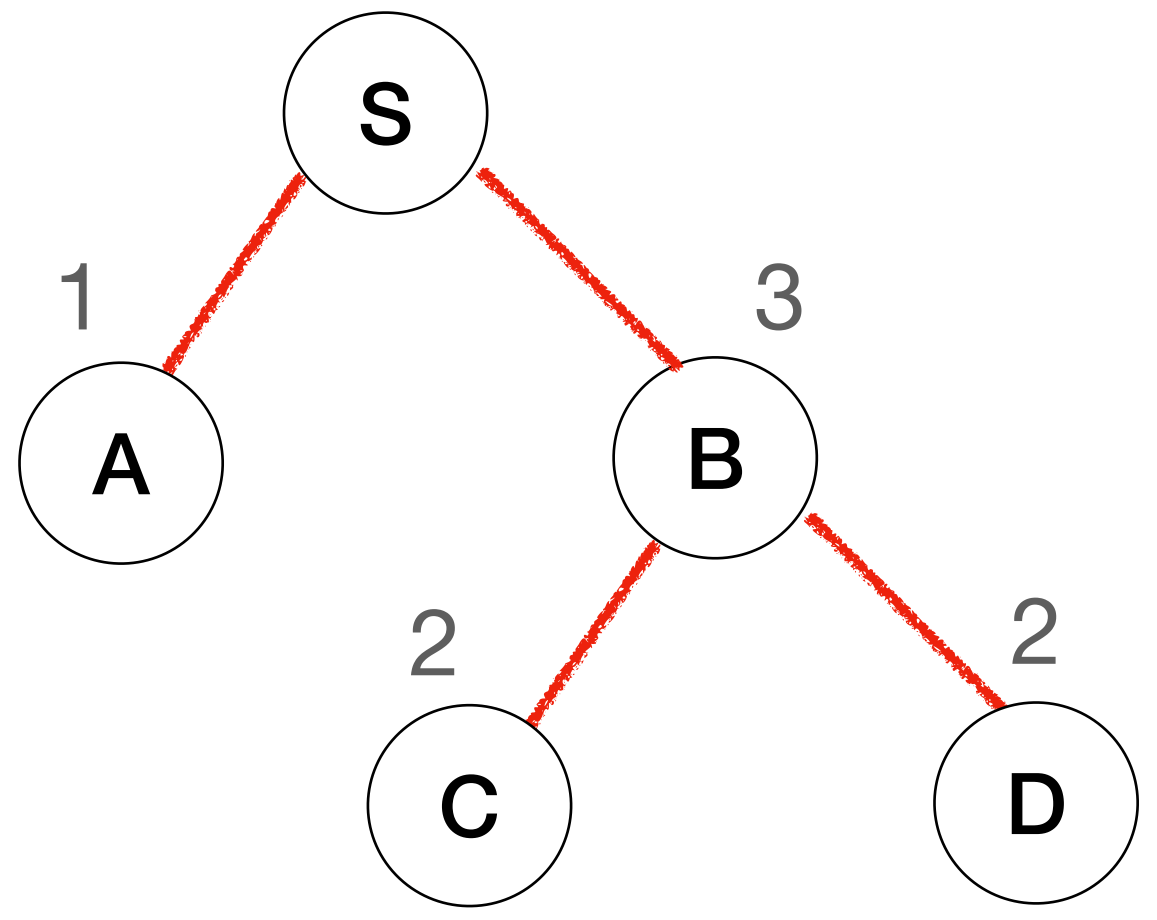
Add vertex B to mstSet

3.mstSet = {S,A,B}



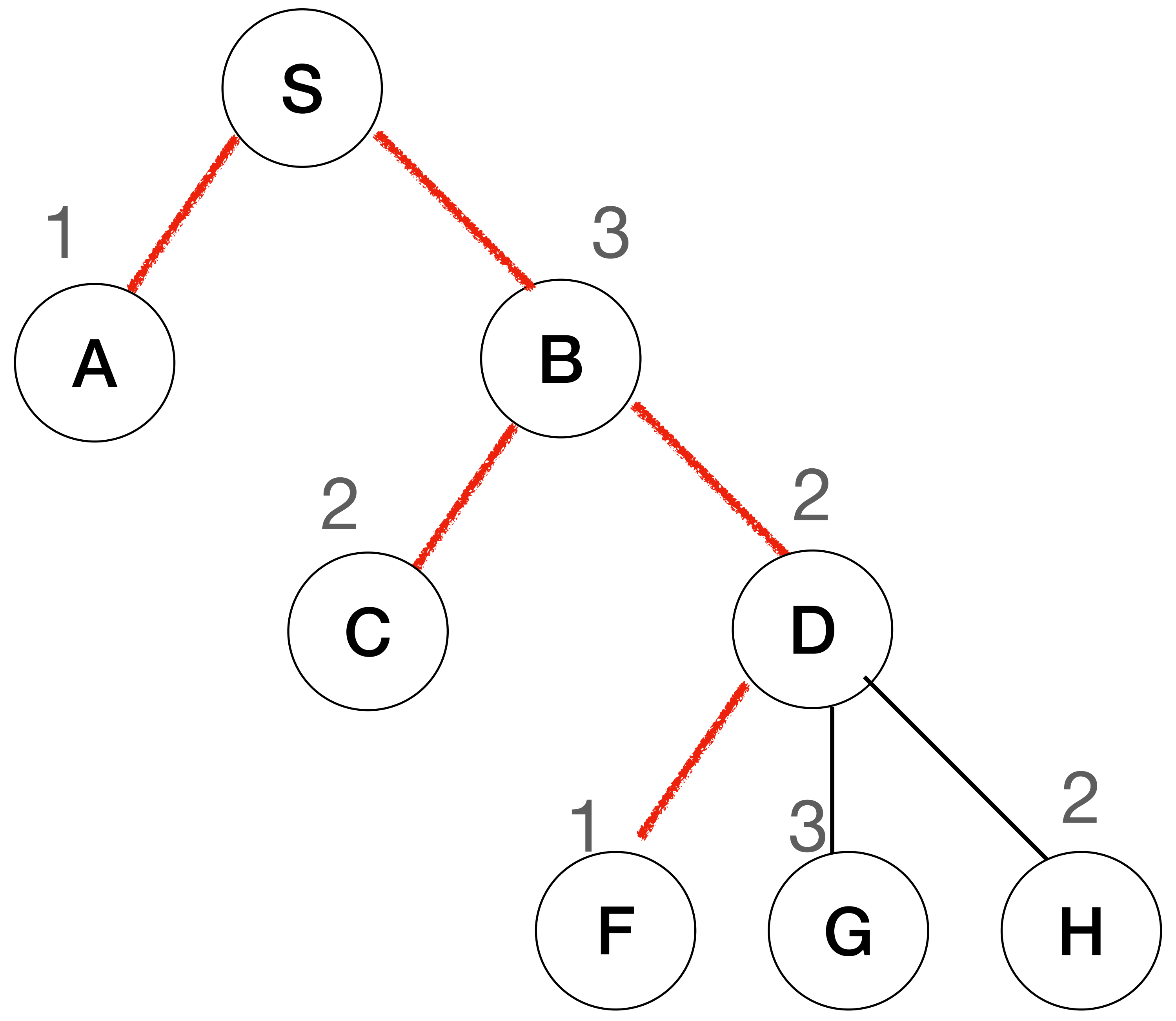
Add vertex C to mstSet

4.mstSet = {S,A,B,C}



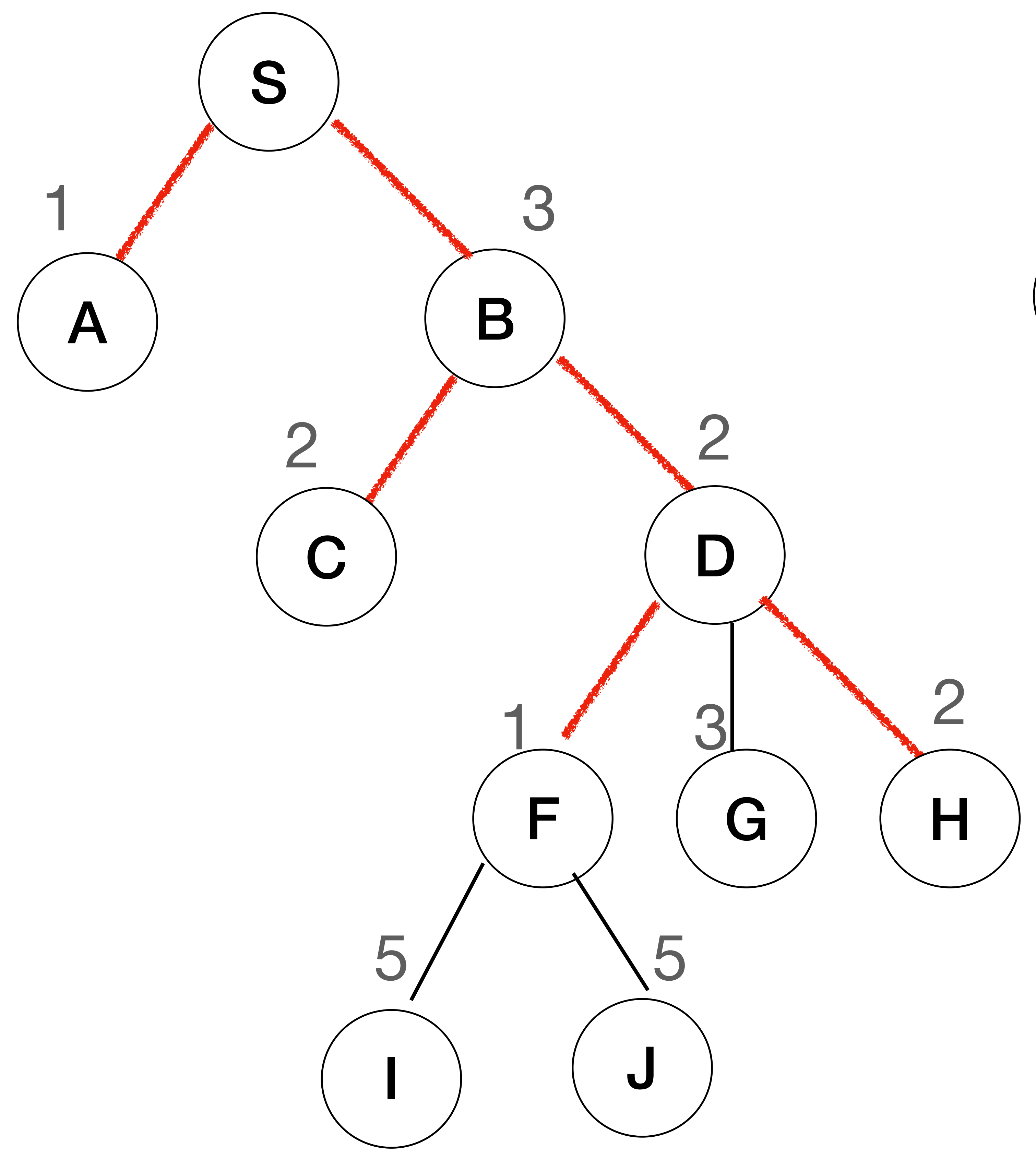
Add vertex D to mstSet

5.mstSet = {S,A,B,C,D}



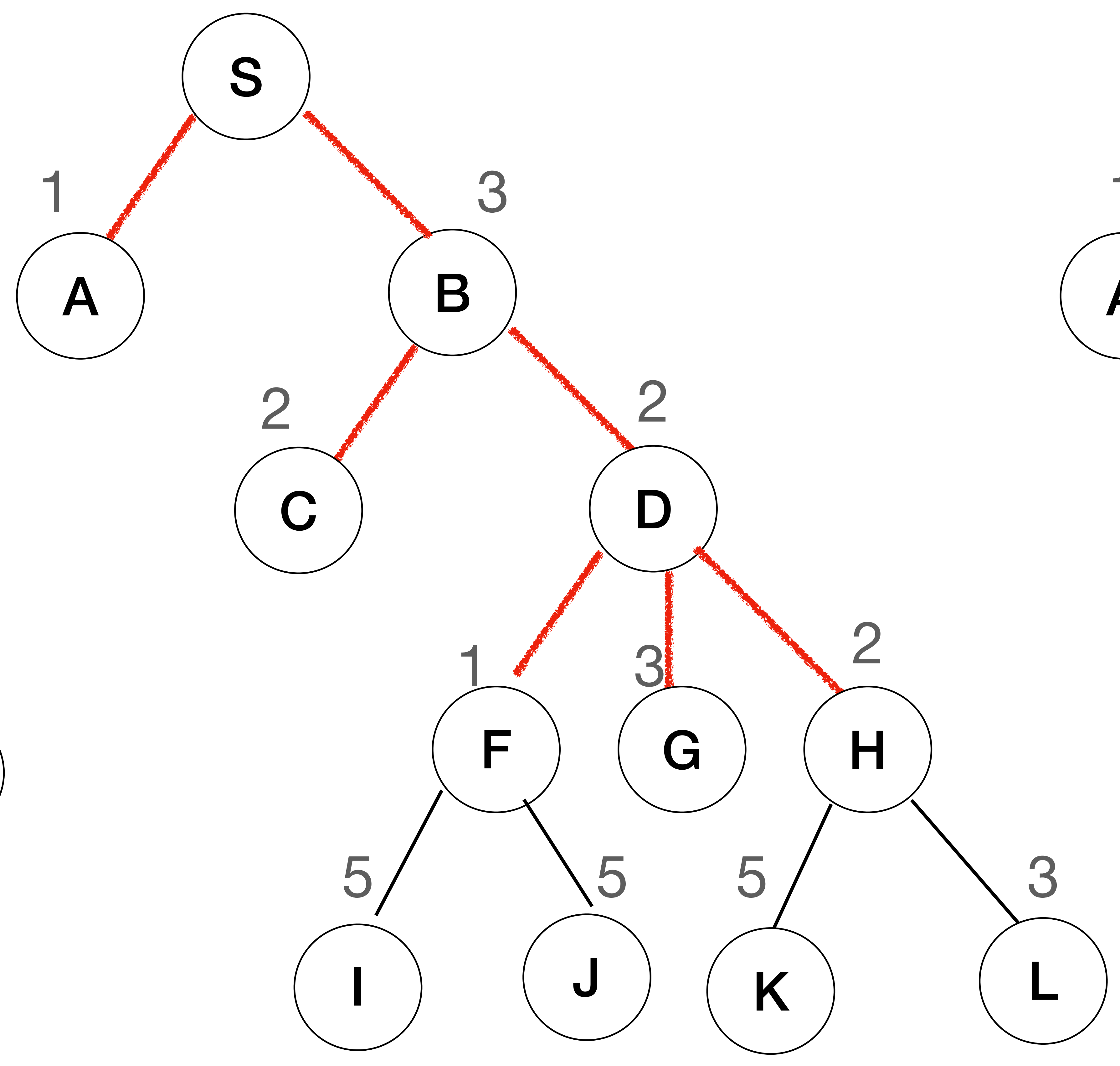
Add vertex F to mstSet

6.mstSet = {S,A,B,C,D,F}



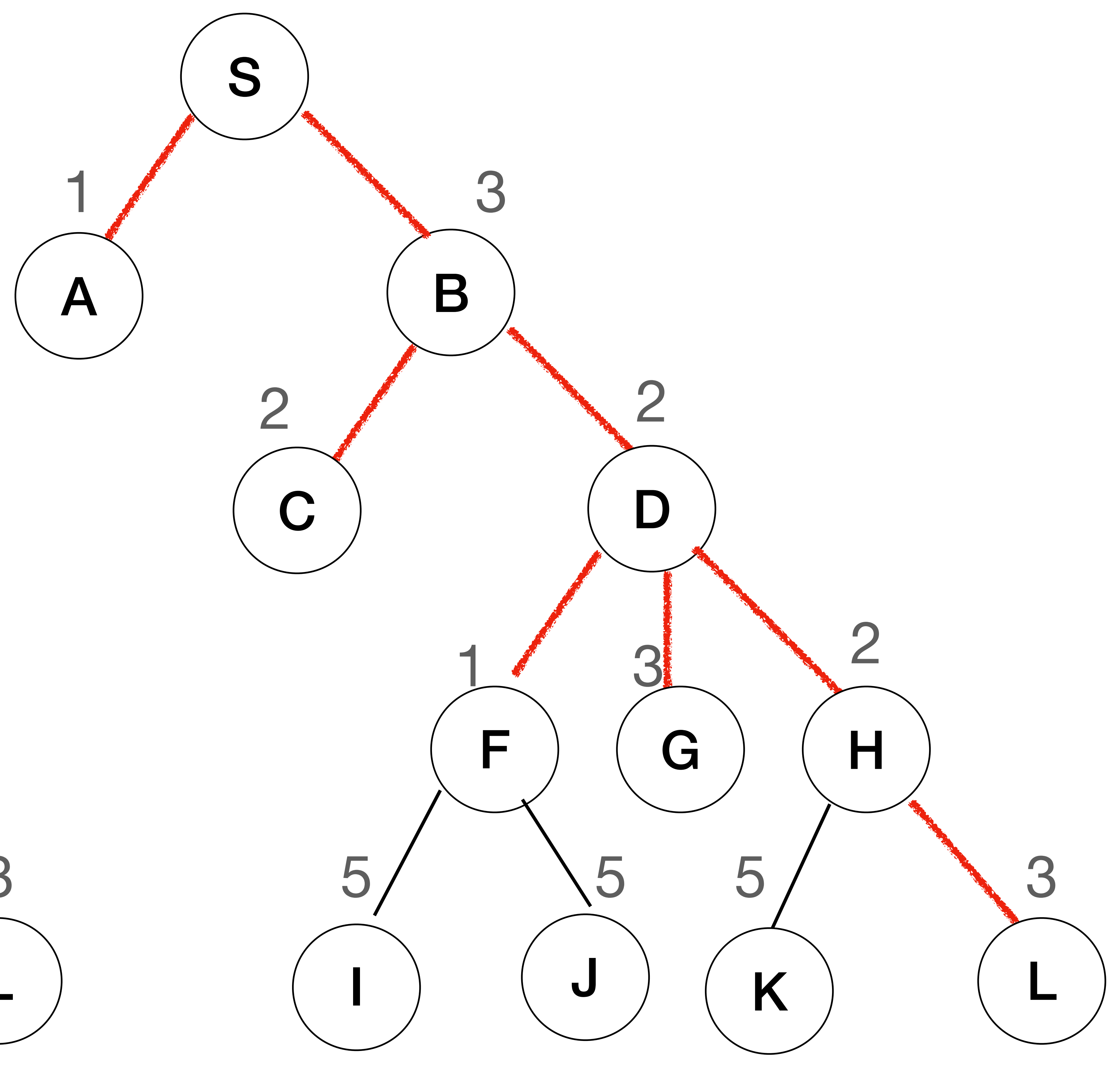
Add vertex H to mstSet

7.mstSet = {S,A,B,C,D,F,H}



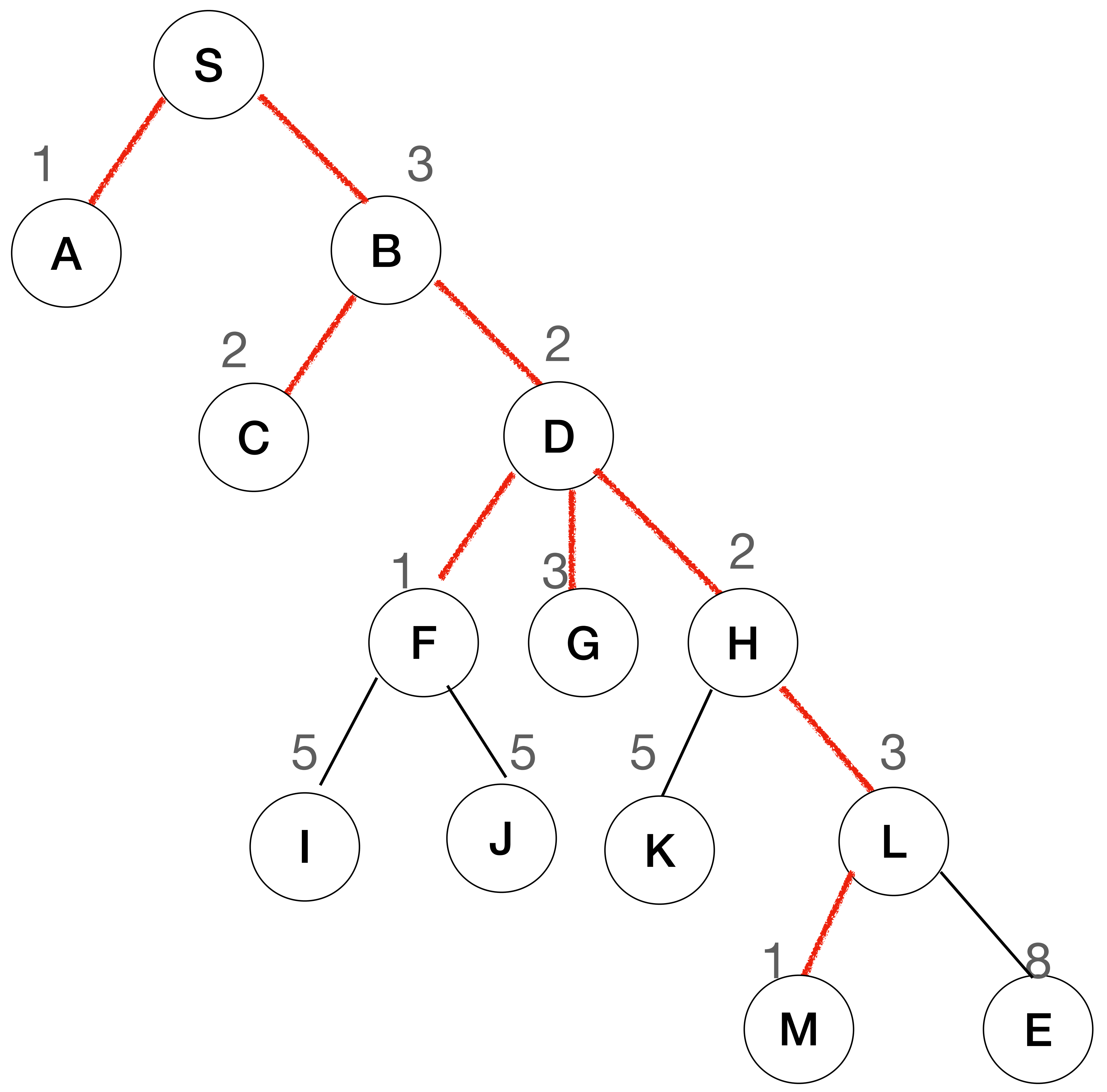
Add vertex G to mstSet

8.mstSet = {S,A,B,C,D,F,H,G}



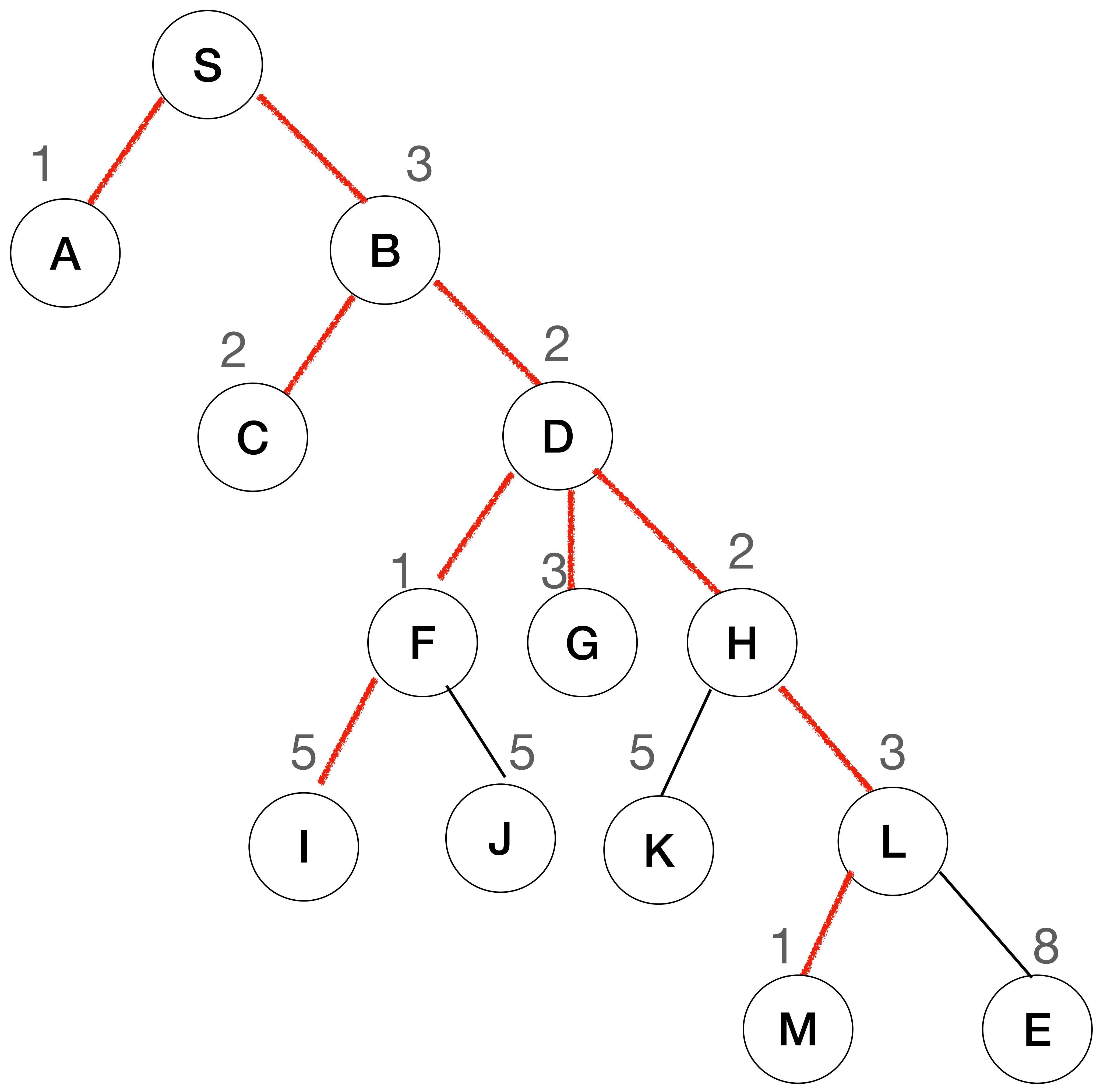
Add vertex L to mstSet

9.mstSet = {S,A,B,C,D,F,H,G,L}



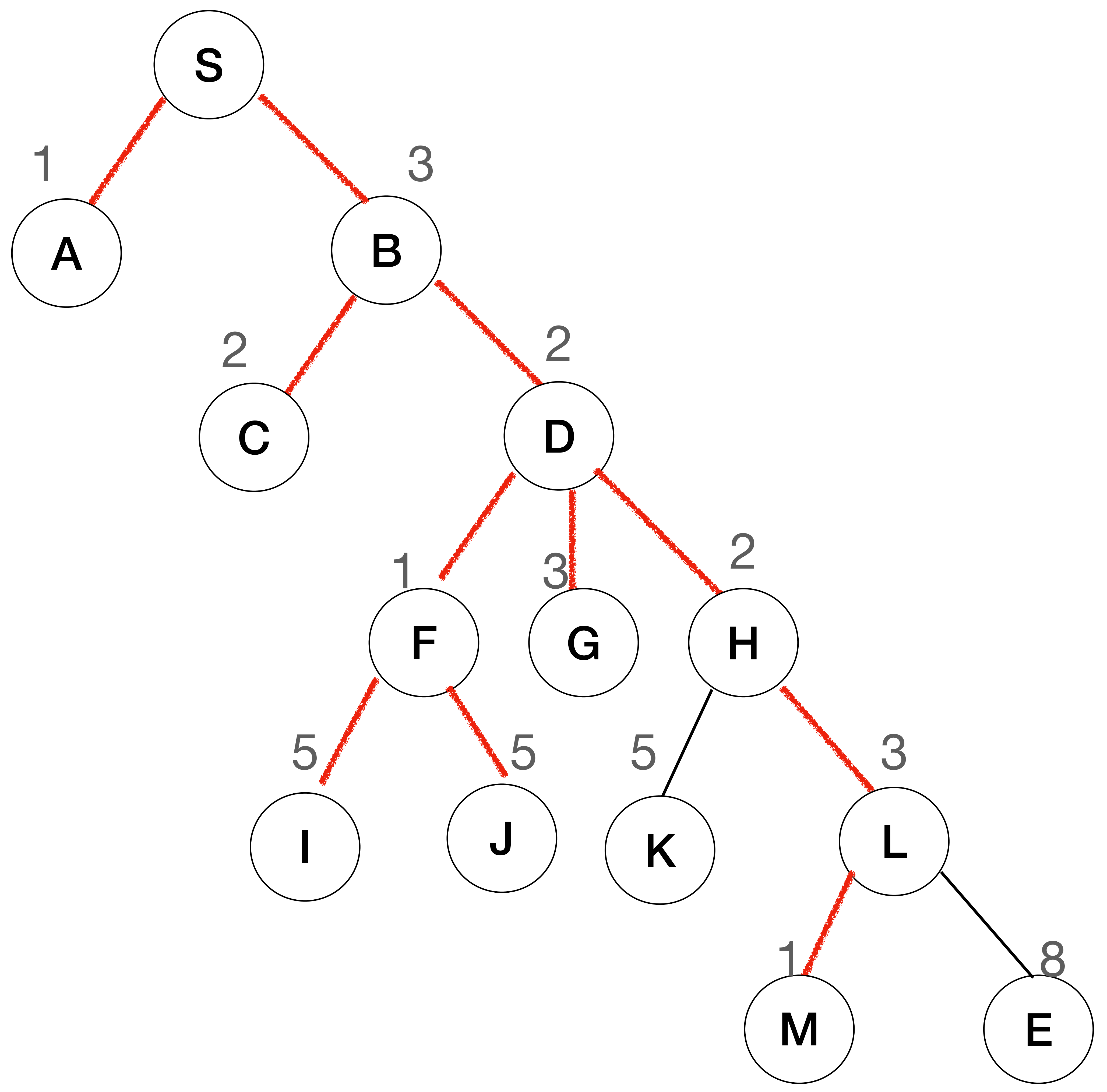
Add vertex M to mstSet

10.mstSet = {S,A,B,C,D,F,H,G,L,M}



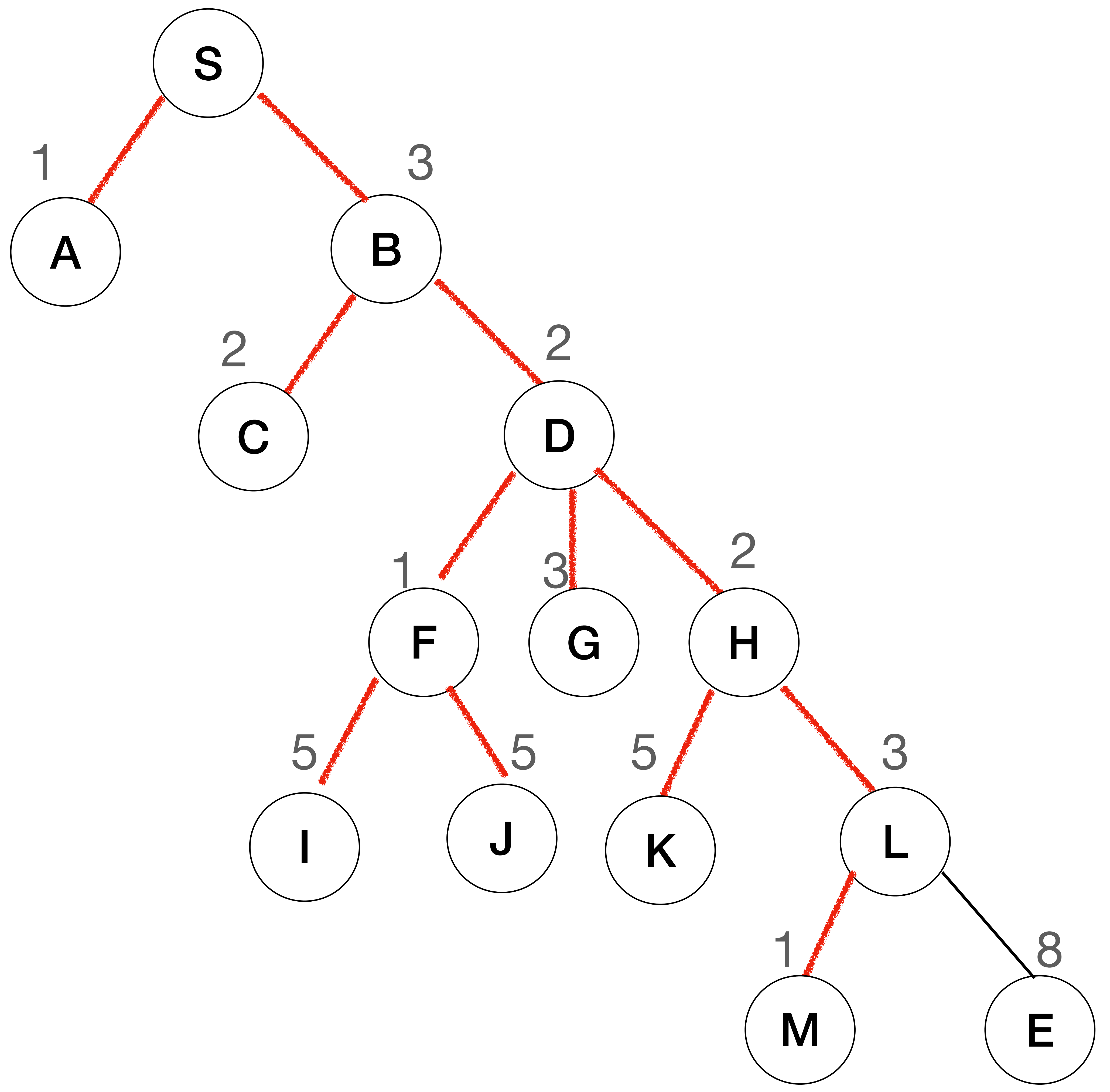
Add vertex I to mstSet

11.mstSet = {S,A,B,C,D,F,H,G,L,M,I}



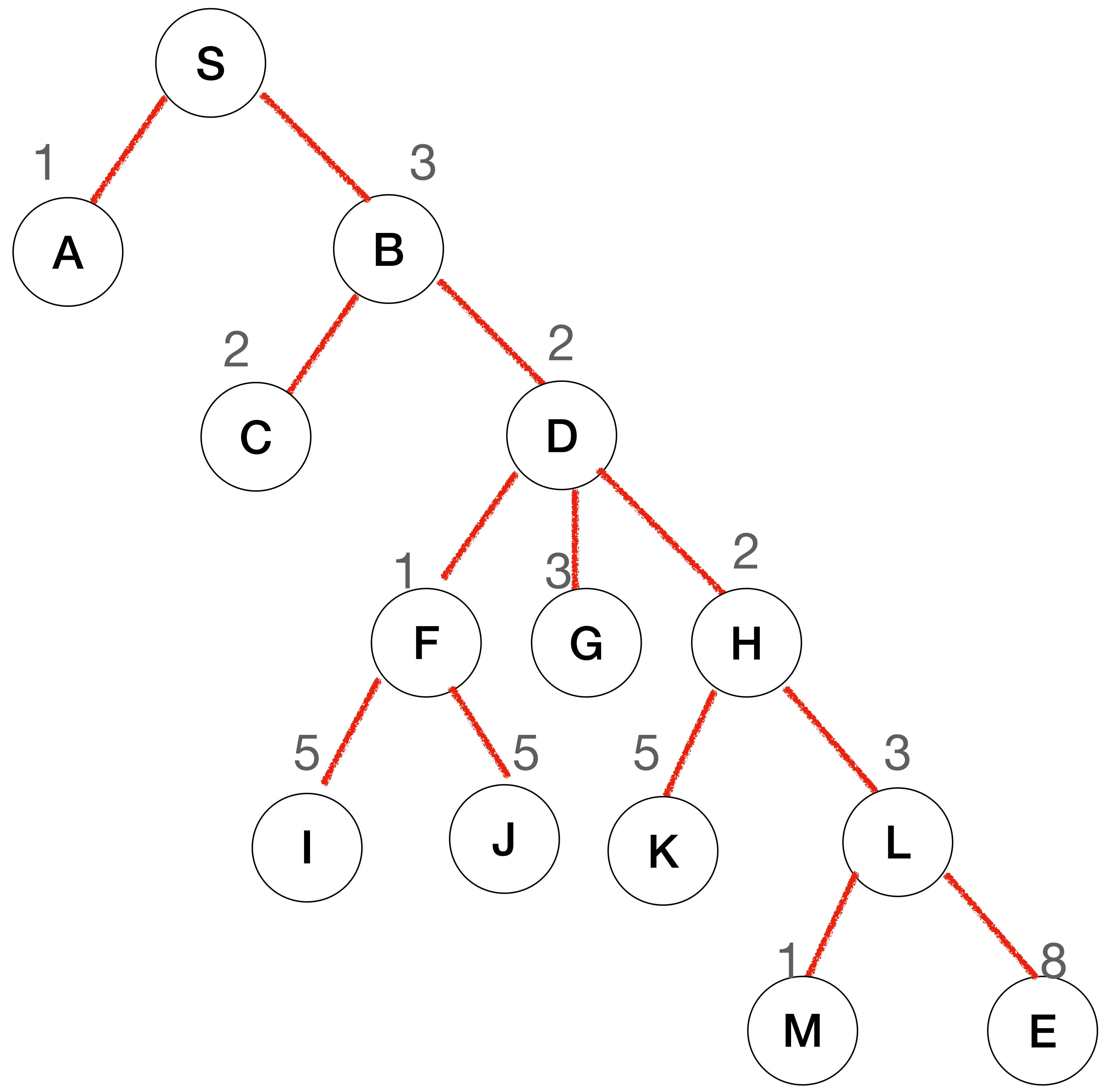
Add vertex J to mstSet

12.mstSet = {S,A,B,C,D,F,H,G,L,M,I,J}



Add vertex K to mstSet

13.mstSet = {S,A,B,C,D,F,H,G,L,M,I,J,K}



Add vertex E to mstSet

14.mstSet = {S,A,B,C,D,F,H,G,L,M,I,J,K,E}

Step 1: Kruskal’s Minimum Spanning Tree algorithm

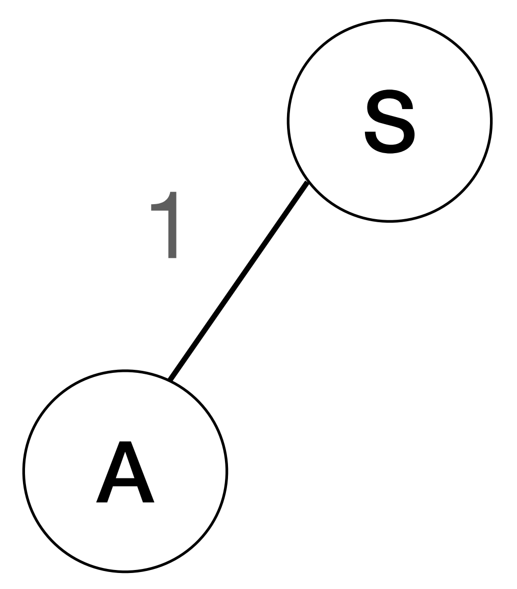
The graph contains 14 vertices and 13 edges. So minimum spanning tree formed will have 13 edges.

After sorting:

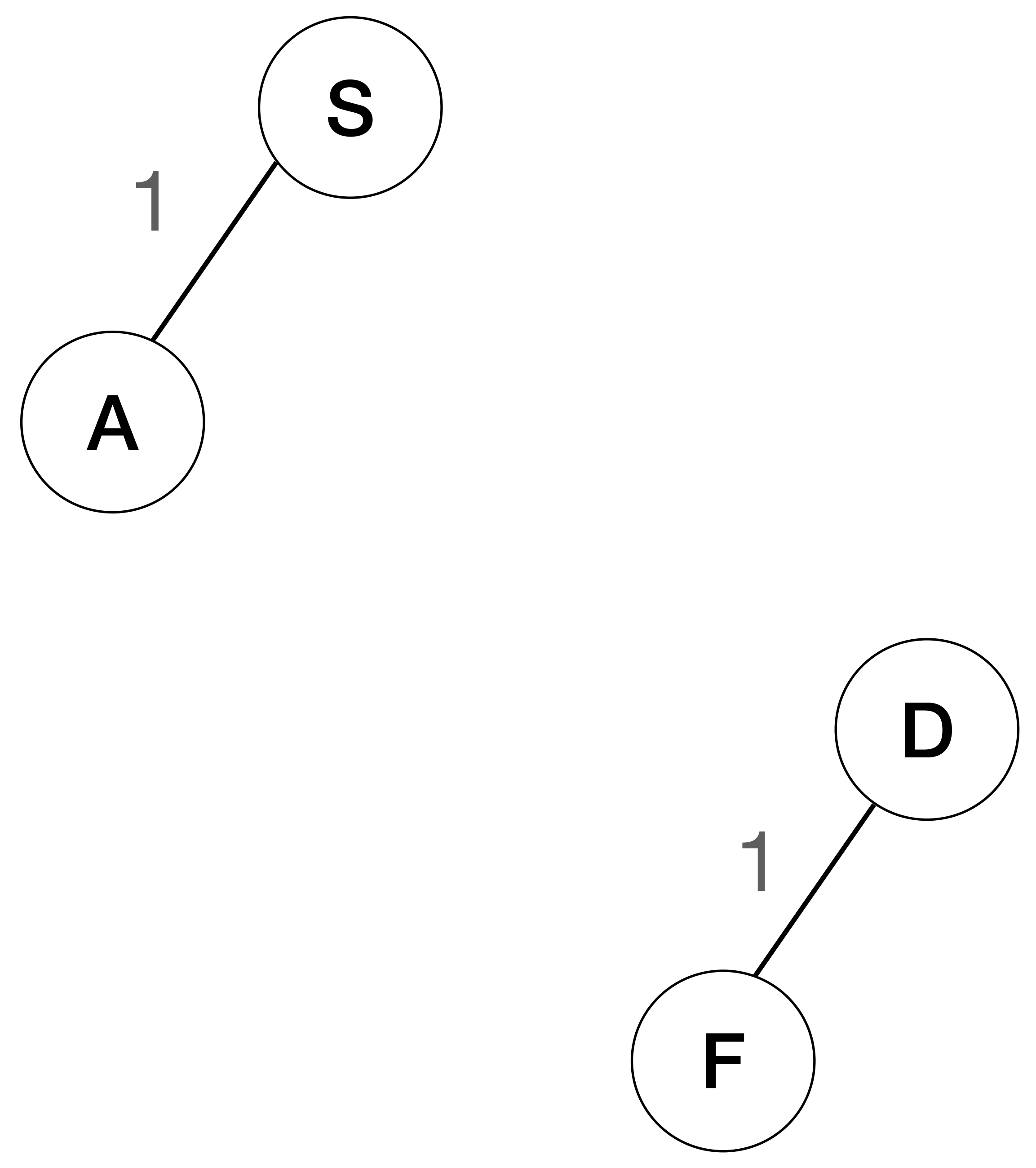
Weight	Source	Destination
1	S	A
1	D	F
1	L	M
2	B	C
2	B	D
2	D	H
3	S	B
3	D	G
3	H	L
5	F	I
5	F	J
5	H	K
8	L	E

Now pick all edges one by one from sorted edges.

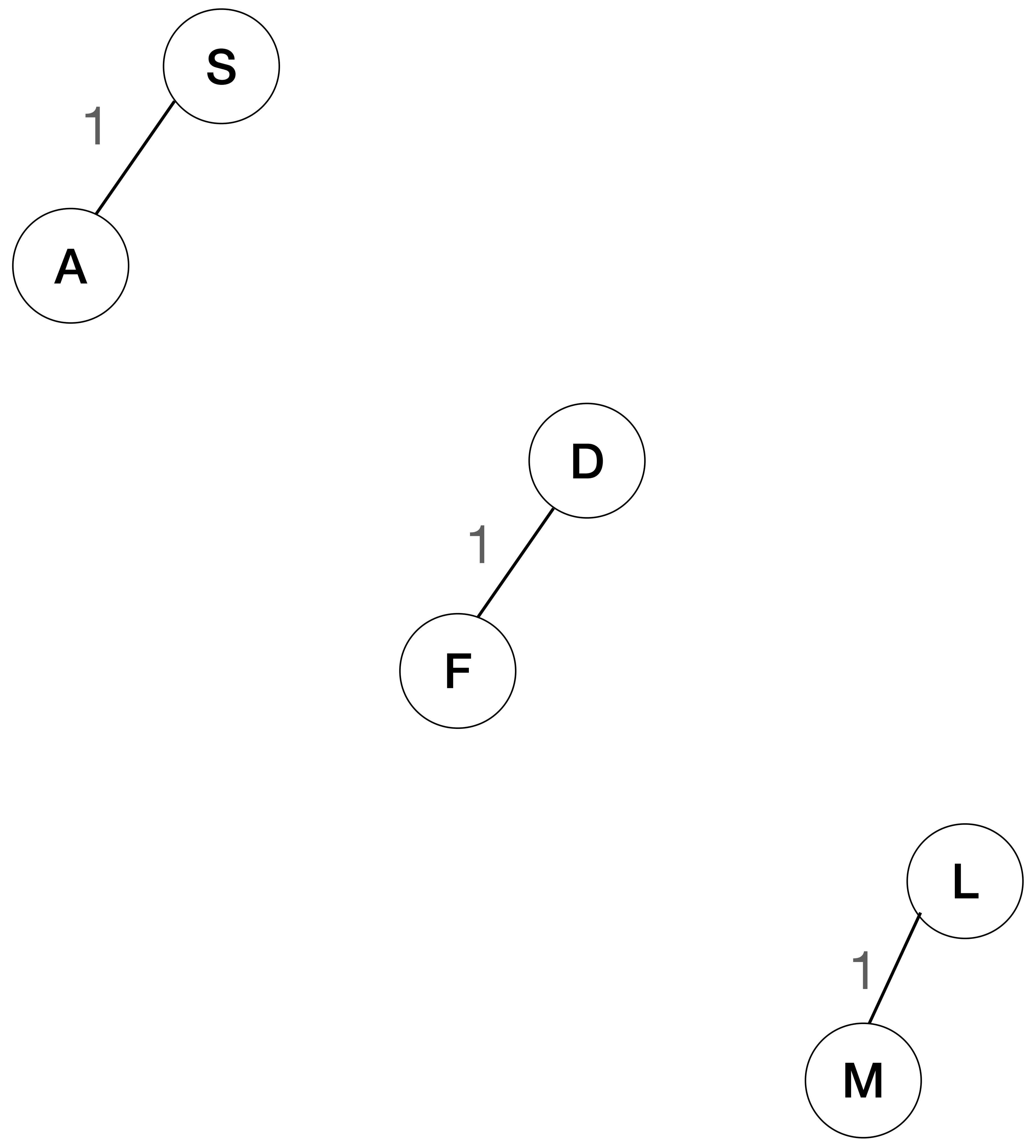
1.Pick edge S-A: No cycle is formed, include it.



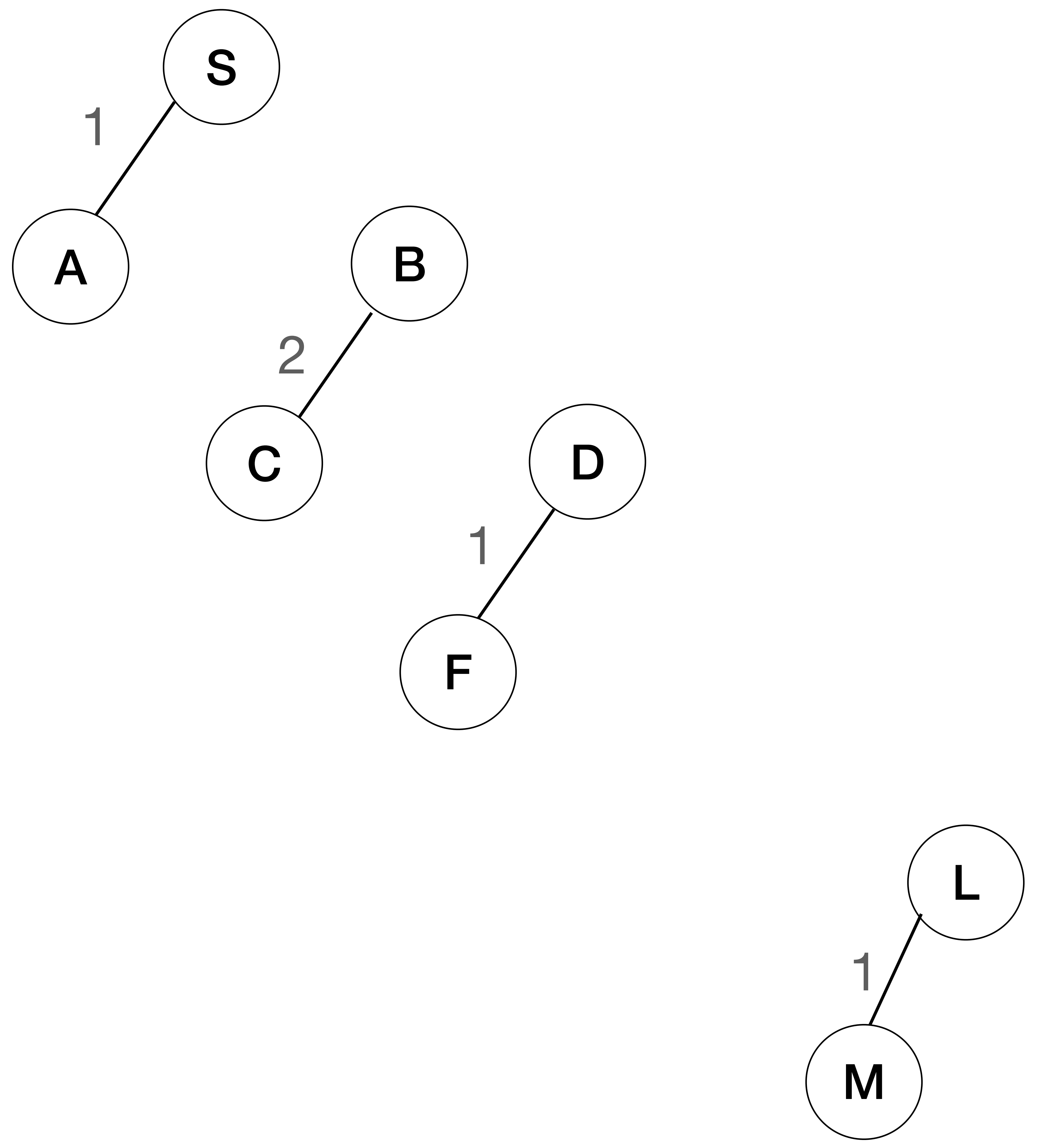
2.Pick edge D-F: No cycle is formed, include it.



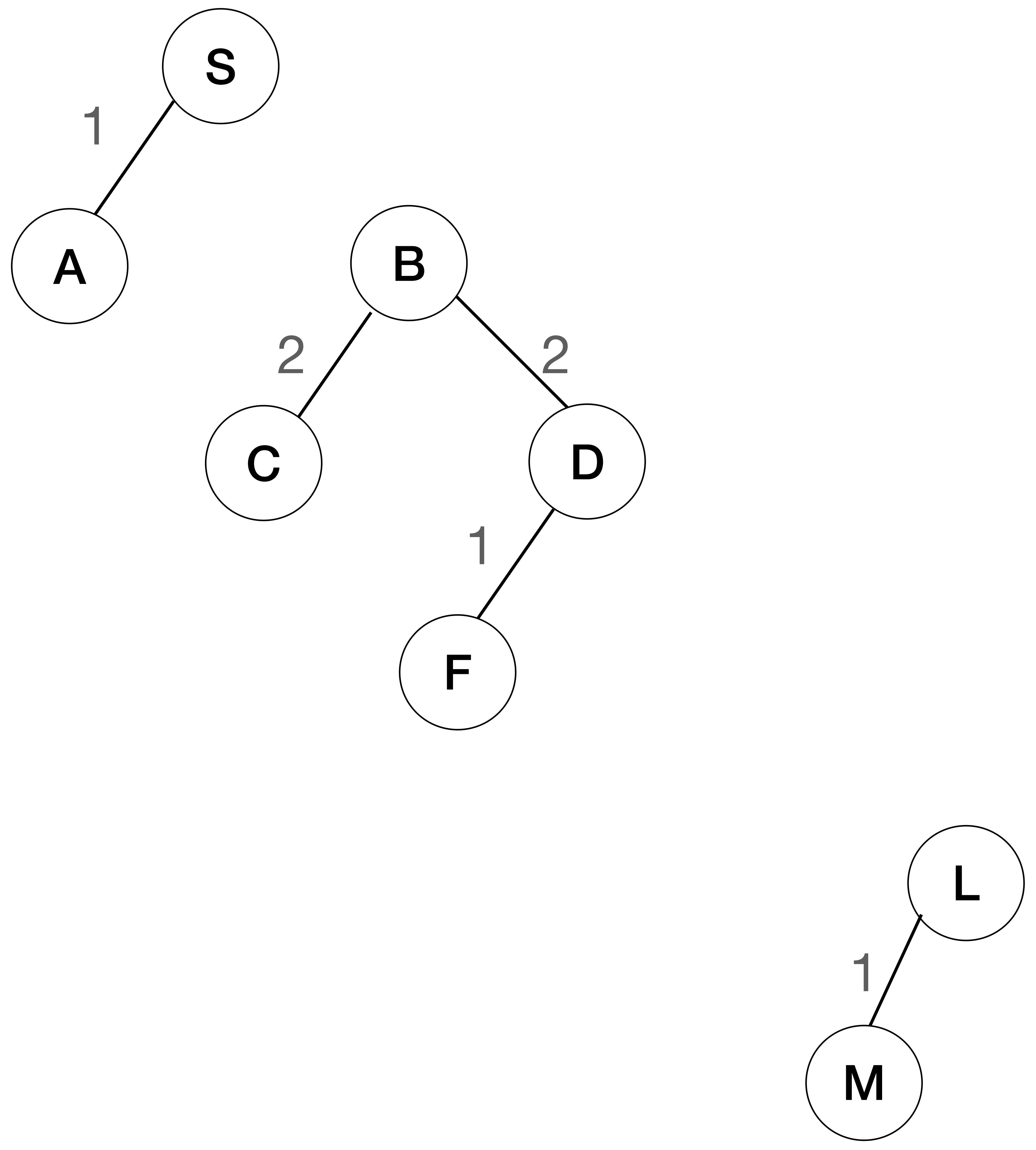
3.Pick edge L-M: No cycle is formed, include it.



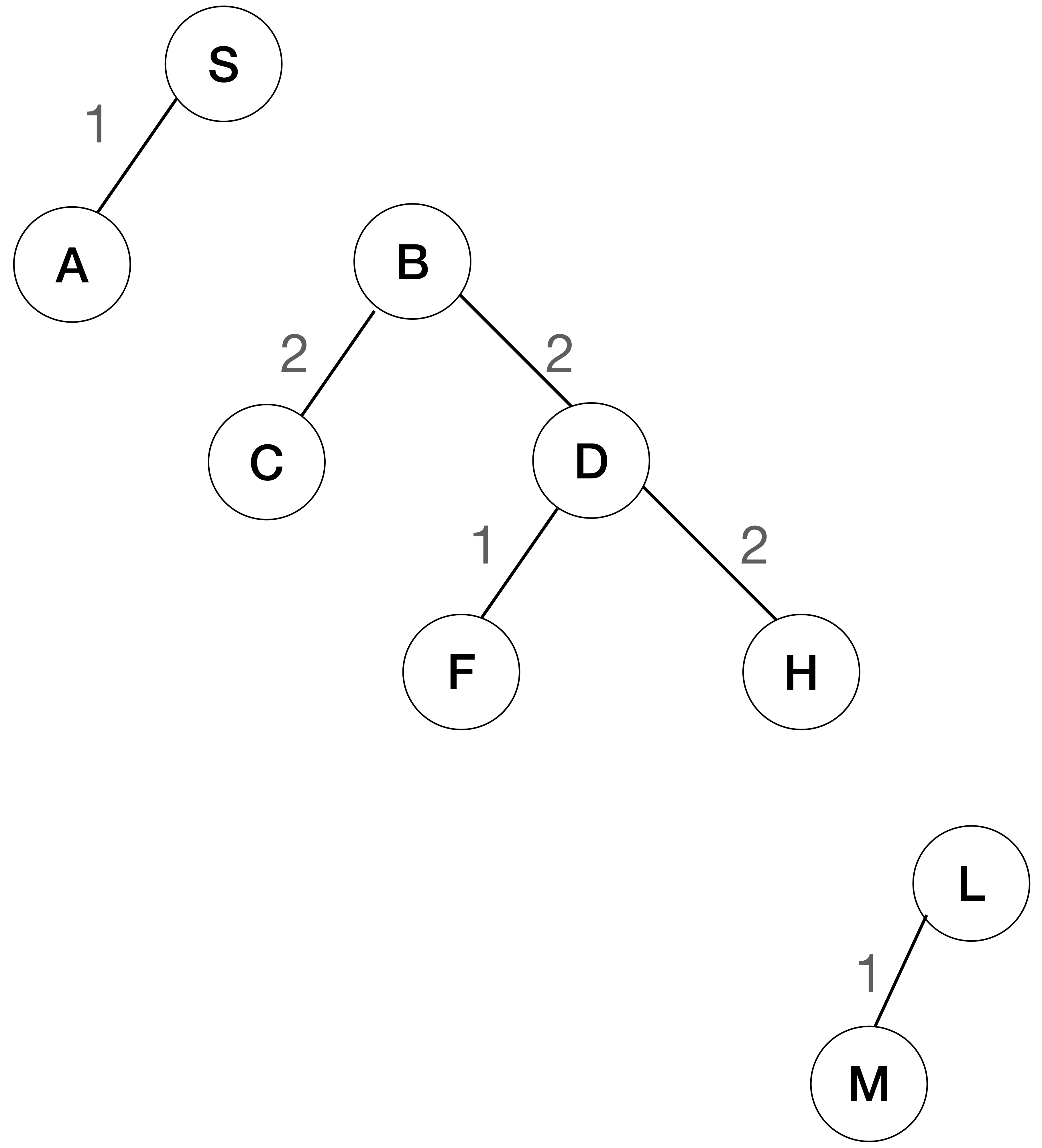
4.Pick edge B-C: No cycle is formed, include it.



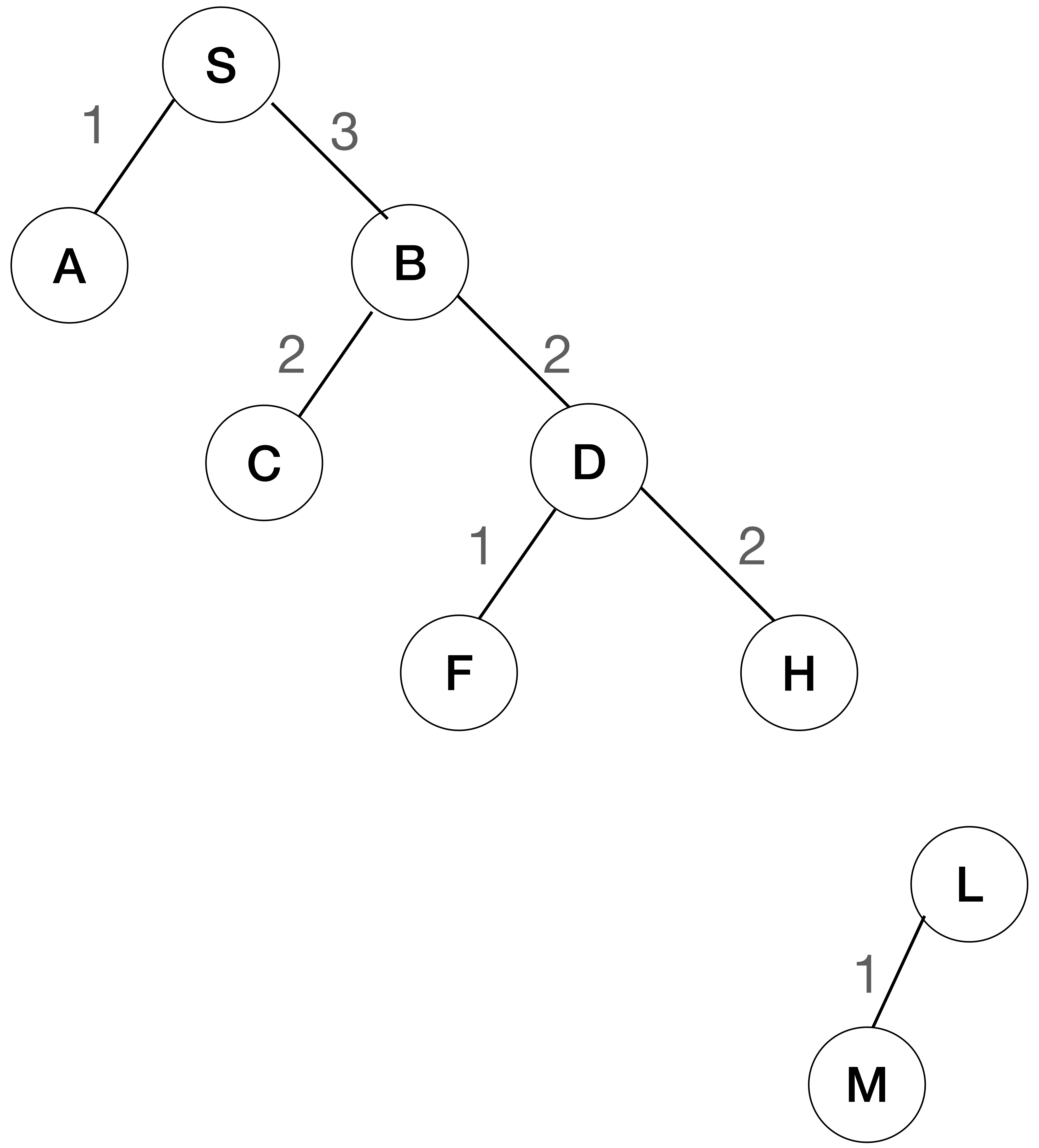
5.Pick edge B-D: No cycle is formed, include it.



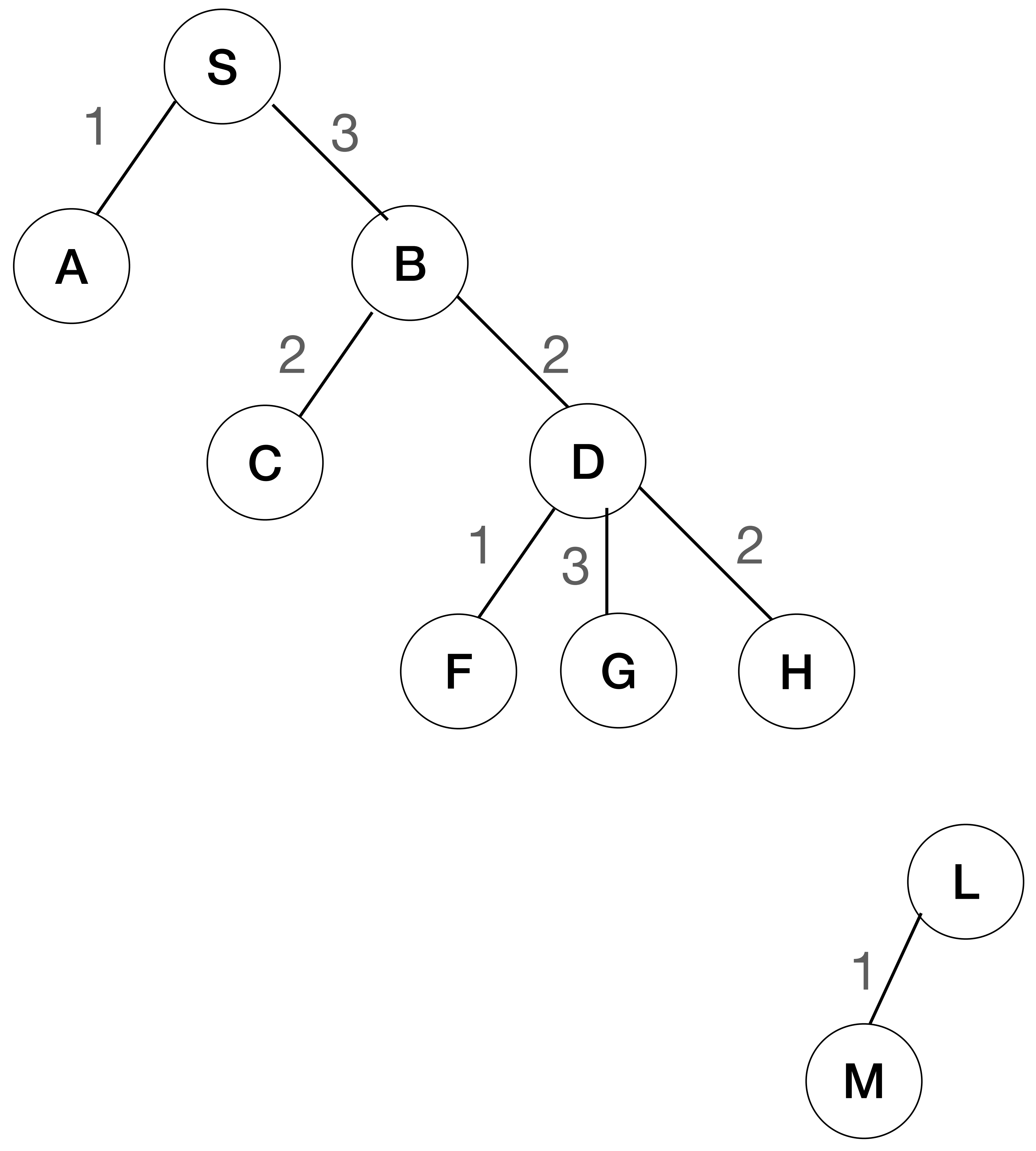
6.Pick edge D-H: No cycle is formed, include it.



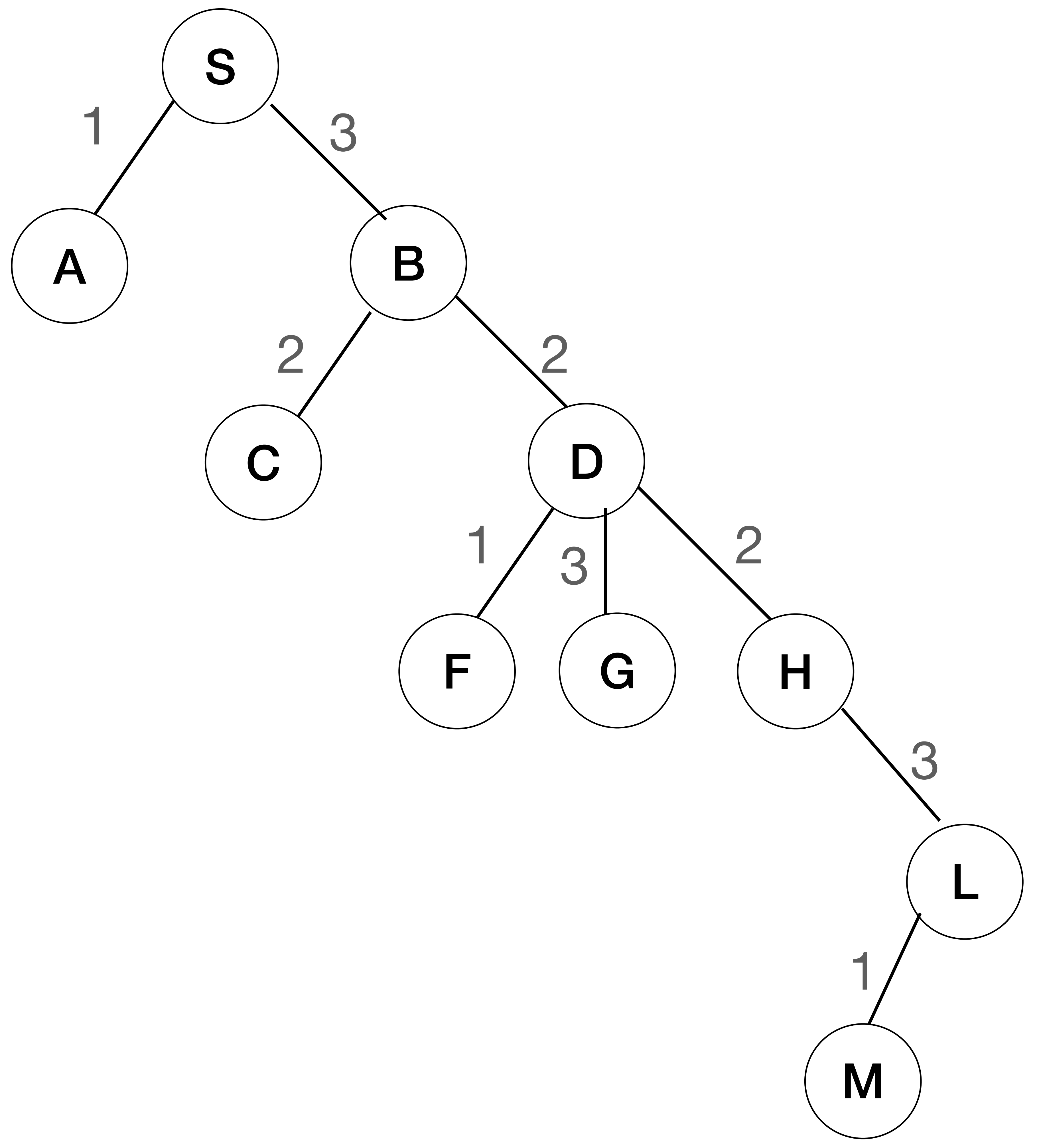
7.Pick edge S-B: No cycle is formed, include it.



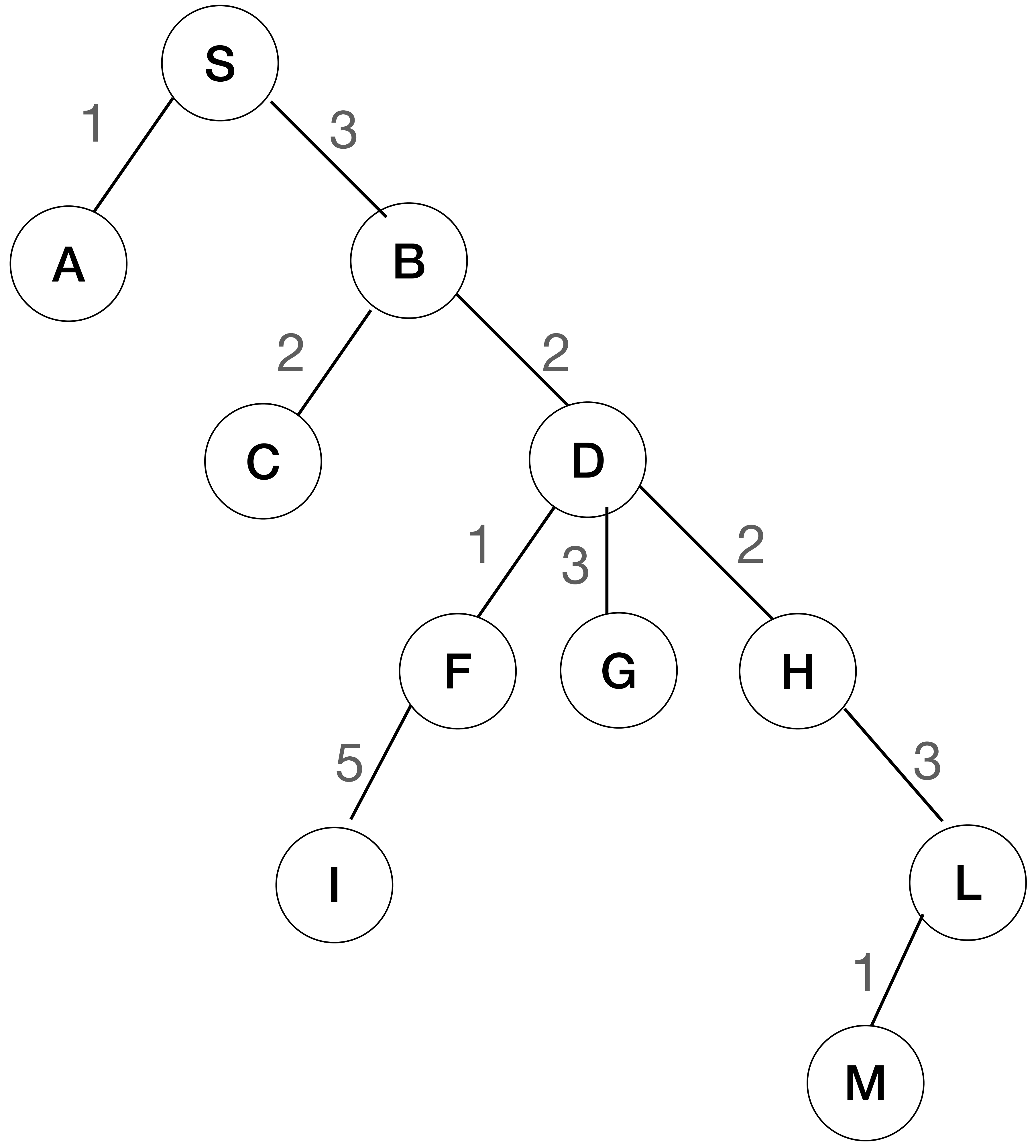
8.Pick edge D-G: No cycle is formed, include it.



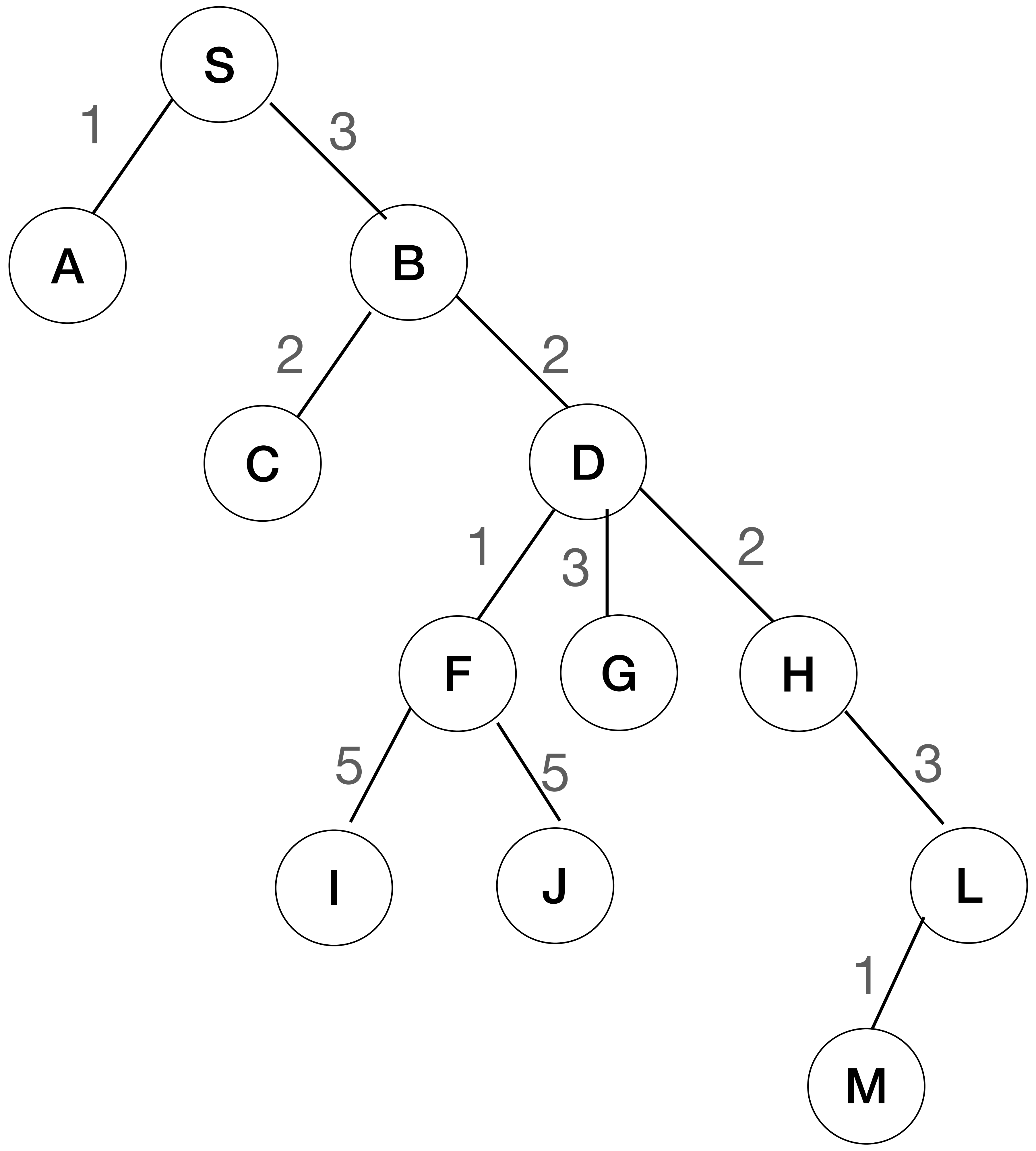
9.Pick edge H-L: No cycle is formed, include it.



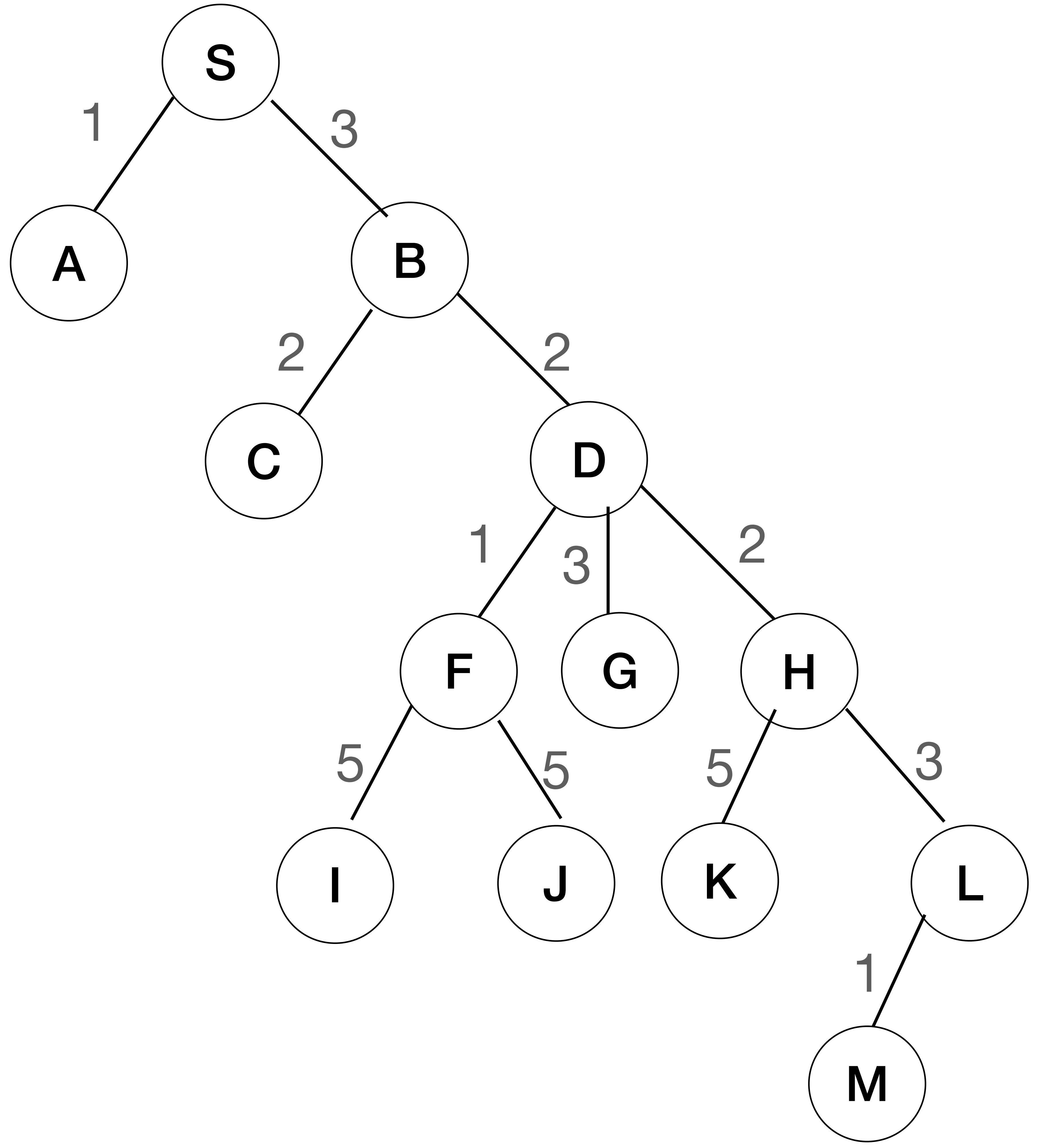
10.Pick edge F-I: No cycle is formed, include it.



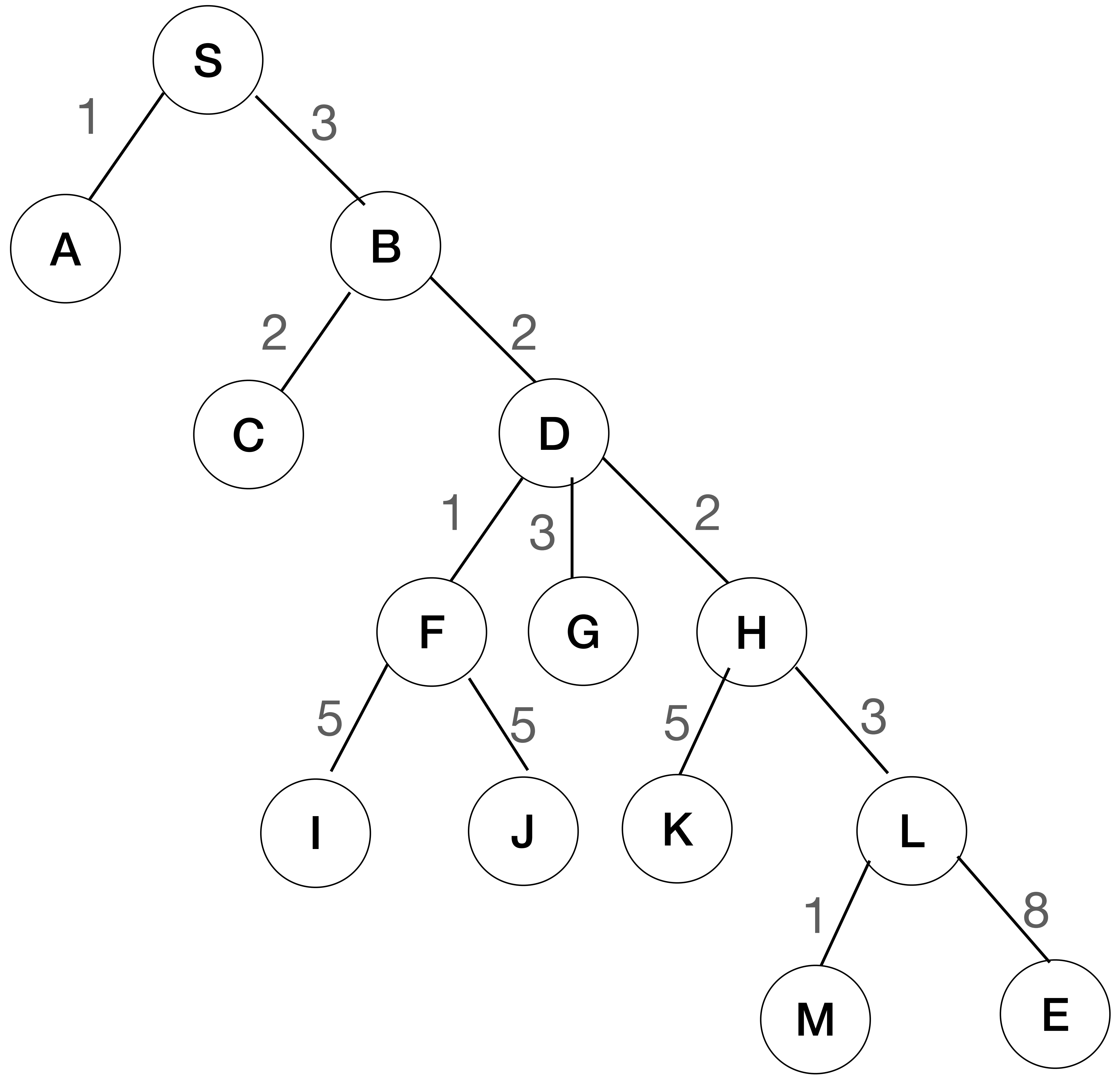
11.Pick edge F-J: No cycle is formed, include it.



12.Pick edge H-K: No cycle is formed, include it.



13.Pick edge L-E: No cycle is formed, include it.



Since the number of edges equals to 13 and all vertices have been visited, the algorithm stops.

Step 2:Big-O comparison

Prim’s MST algorithm: O((V + E) * logV), V is the number of vertices. Each vertex is inserted in the priority queue only once and the time spent on insertion is logarithmic time. It runs faster in dense graphs.

Kruskal’s MST algorithm: O(E * logV),V is the number of vertices. Most time is spent on sorting. It runs faster in sparse graphs.