CS 521
Quiz 2

1. Given a source vertex *s* in an undirected graph G the distance d(s,u) from *s* to a vertex u is defined as the minimum number of edges in any path from s to u. The distance d(s,u) =∞ if there is no path from *s* to u. Answer true or false for the following claim: For any edge (u,v) in G, $d(s,v) \leq d(s,u) + 1$.

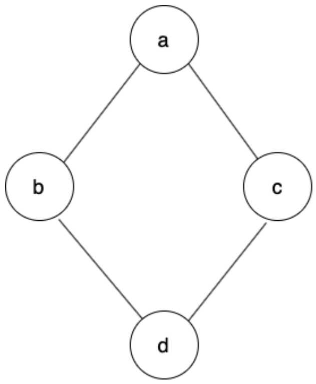
   Answer: True

   Proof: Suppose not, i.e., $d(s,v) > d(s,u) + 1$. But there is a path from s to v that goes from s to u with d(s,u) edges followed by the edge (u,v). Contradiction.

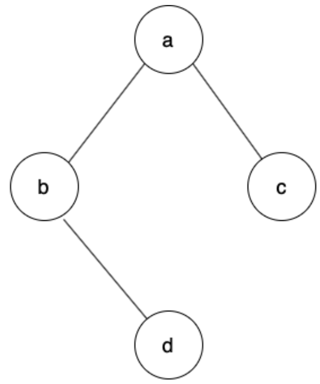
2. Answer true or false for the following claim: The breadth-first search tree produced by the BFS algorithm does not depend on the order in which the edges are listed in the adjacency lists for the input graph. (Note that two trees T_1 and T_2 are considered different if, for some i, the subtrees rooted at the i^{th} child of the root in each tree are different, i.e., the left-to-right order of the subtrees matters.)

   Answer: False

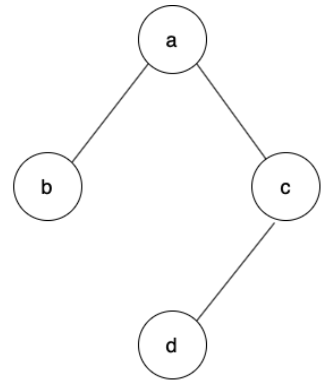
   Counterexample



Graph G



BFS tree **T** where (a-b) is listed first



BFS tree **T'** where (a-c) is listed first

3. Given a complete graph on n vertices and a source vertex s, the depth-first search tree produced by the DFS algorithm will always have a single leaf vertex.

   True

   Proof: By induction on n.

4. In any depth-first search tree produced by the DFS algorithm, for any two distinct vertices u and v, neither u nor v is a descendant of the other if and only if in the recursive invocations:
   a. Either DFS(u) was called before DFS(v) but DFS(v) returned before DFS(u), or DFS(v) was called before DFS(u) but DFS(u) returned before DFS(v).
   b. DFS(u) was called after DFS(v) returned.
   c. DFS(v) was called after DFS(u) returned.
   d. Either DFS(u) was called after DFS(v) returned, or DFS(v) was called after DFS(u) returned.

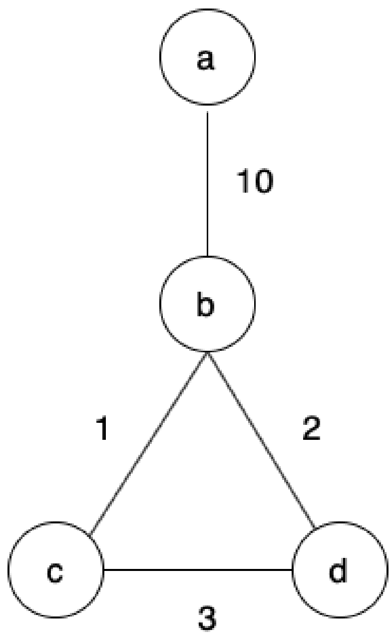
   Answer: d

   Proof: This follows from the property that v is a descendant of u in the DFS tree iff DFS(v) is called after DFS(u) is invoked and before DFS(u) returns.

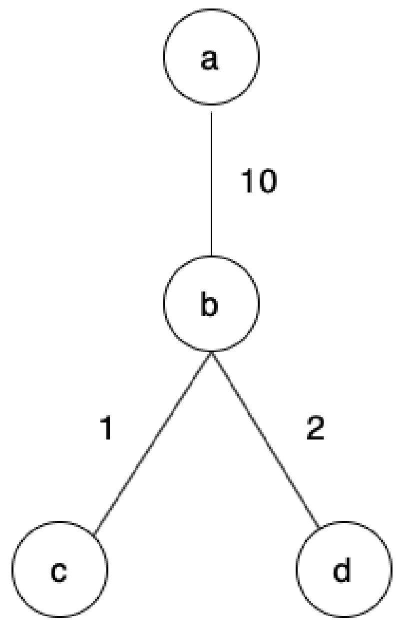
5. Let G be any connected undirected graph with a distinct cost for each edge. Let e be the edge with the maximum cost. Answer true or false for the following claim: The edge e can never appear in any minimum spanning tree for G.

   Answer: False

   Counterexample



: Graph G with e:(a,b)

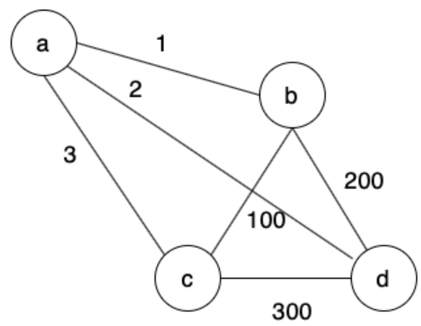


MST **T** contains e:(a,b)

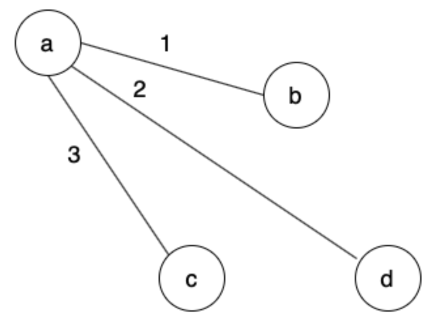
6. Let G be any connected undirected graph with a distinct cost for each edge, and e the cheapest edge in some cycle in G. Answer true or false for the following claim: The minimum spanning tree of G contains e.

   Answer: False

   Counterexample:



Graph G with e:(b,c), the
minimum weight edge in the cycle b-c-d-b



MST **T** does not contain e:(b,c)

7. Let G be any connected undirected graph with a distinct cost for each edge. Answer true or false for the following claim: If the edge e=(u,v) appears in the minimum spanning tree for G then there is no path in G from u to v that consists entirely of edges that are all cheaper than e.

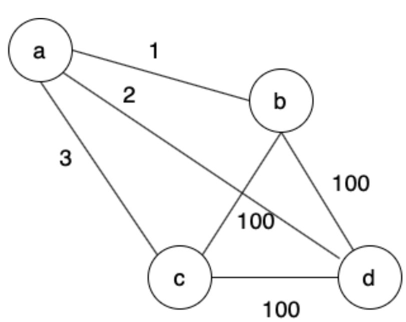
   Answer: True

   Proof: This is just a restatement of the Cycle Property.

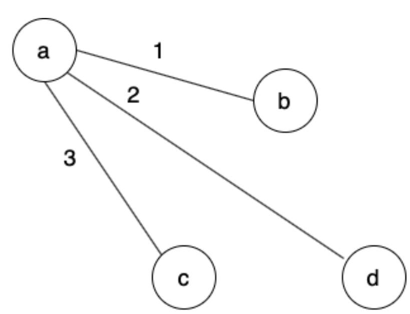
8. Let G be any connected undirected graph with a unique (i.e., a single) minimum spanning tree. Answer true or false for the following claim: All the edges weights in G are distinct.

   Answer: False

   Counterexample:



Graph **G** with not all edges are of distinct weight



MST **T** is Unique

9. Answer true or false for the following claim: Let G be any directed acyclic graph which has a topological ordering with v_1 the first

node and v_n the last node. Then there must be a directed path in G from v_1 to v_n.

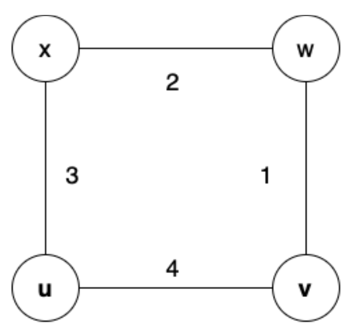
Answer: False

Counterexample: Suppose the nodes in G are u, v, s and t and the only edges are (u,v) and (s,t). Then u,v,s,t is a topological ordering, but there is no directed path in G from u to t.

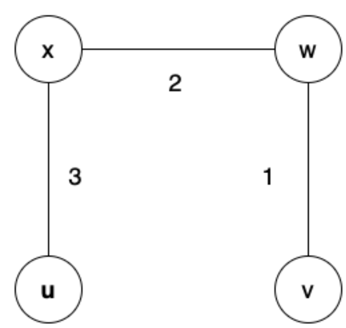
10. Let G be any connected undirected graph with a distinct cost for each edge. Answer true or false for the following claim: The path between any pair of vertices u and v in the minimum spanning tree of G is a shortest path between u and v in G.

Answer: False

Counterexample:



Graph **G** with all edges are of distinct weight.



Shortest path between **u** and **v** in MST **T** is not the same as that in **G**