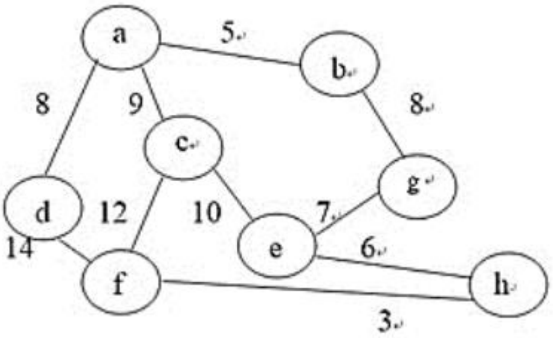
# Algorithm 2015
# fall Homework 4

1. Prove the edge with the second smallest weight must be in a minimum spanning tree.
2. Give an O(|V|) time algorithm for determining whether or not a given undirected graph G (V, E) contains a cycle. Explain why your algorithm's running time is independent of |E|.
3. For the following graph,
   A. What is a minimum-cost spanning tree?
   B. Write an algorithm to describe how you get the result of (A).



4. Let (u,v) be a minimum-weight edge in a connected graph G. Show that (u,v) belongs to some minimum spanning tree of G.
5. Give an algorithm that determines whether or not a given undirected graph G = (V ,E) contains a cycle. Your algorithm should run in O(V) time, independent of |E|.
6. The topological sort of an arbitrary directed graph G = (V, E) can be computed in linear time. (True or False)
7. Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected

activities. Describe how this approach is a greedy algorithm, and prove that it yields an optimal solution.

8. Find the strongly connected components on a directed graph in Figure 1.

9. Is a directed graph in Figure 1 acyclic? (Hint: Use the depth-first-search algorithm DFS.)

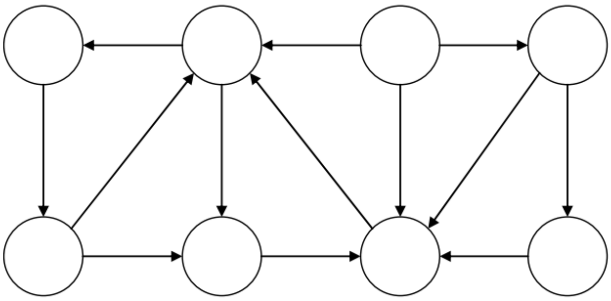


Figure 1: A directed graph.

10. Using the depth-first-search algorithm DFS on an undirected graph in Figure 2. Vertices are timestamped by discovery time/finishing time. (Hint: Ignore the weight of each edge.)

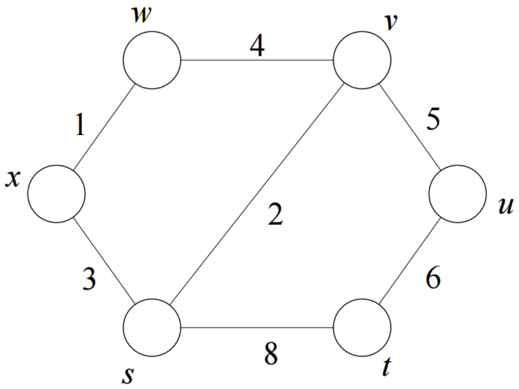


Figure 2: An undirected graph.