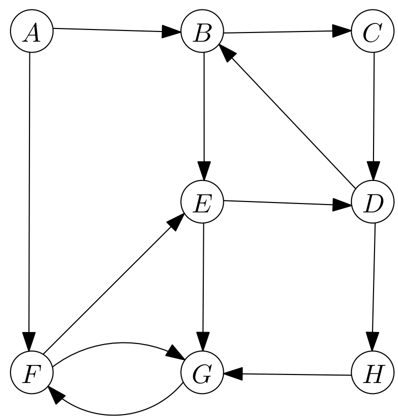
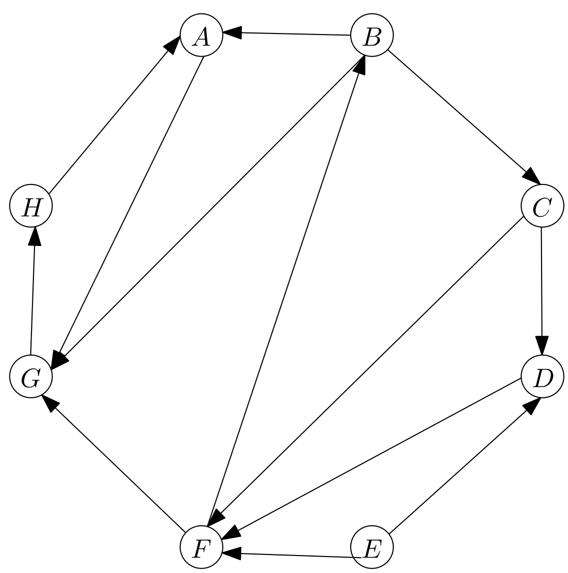
Chapter 3

——Material covered in class——

1. Perform depth-first search on each of the following graphs; whenever there is a choice of vertices, pick the one that is alphabetically first. Classify each edge as a tree edge, forward edge, back edge, or cross edge, and give the pre and post number of each vertex.

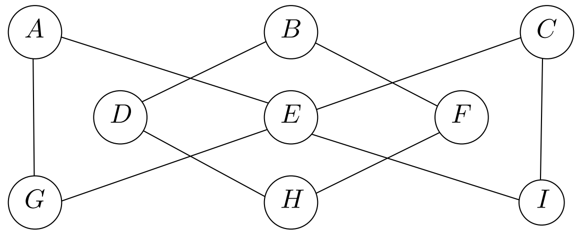


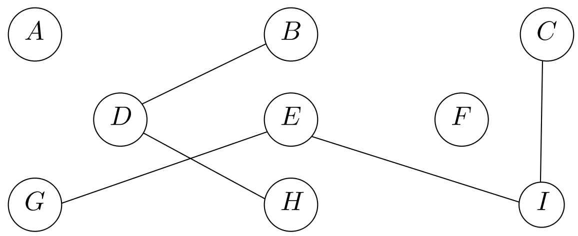
(a)



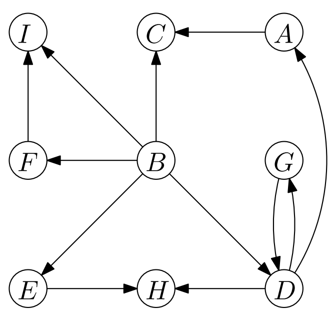
(b)

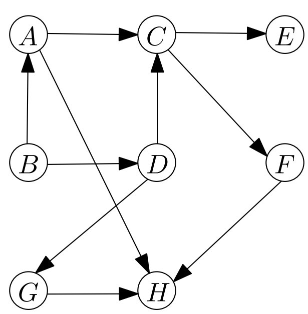
2. For each of the following two graphs, find all connected components by doing the trace of the algorithm we studied in class.

(a)


(b)


3. In class, we studied an algorithm which can detect if a directed graph is cyclic or acyclic. Show the trace of this algorithm on the following inputs.

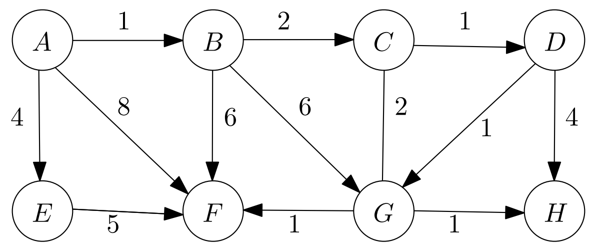
(a)




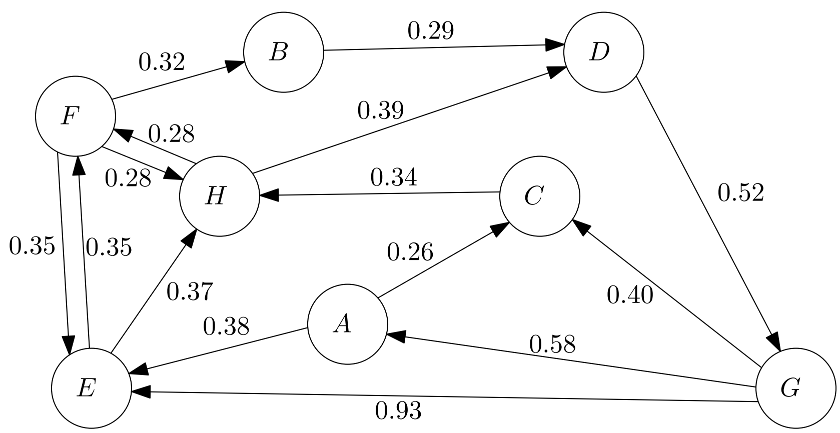
(b)

4. For each graph in Question 3, say whether it has a topological sort or not. If the graph has a topological sort, give one. If not, say why.

5. Show the trace of Dijkstra's algorithm on the following inputs, starting at node A.



(a)



(b)

6. In Question 5, show the state of the min-heap at each step of your traces.

—Design & analysis of algorithms—

7. Modify the pseudocode for depth-first search so that it prints out every edge in the directed graph $G = (V, E)$, together with its type. Show what modifications, if any, you need to make if G is undirected.

8. Write the pseudo-code for breadth-first search (BFS). What is the running time of breadth-first search on a graph $G = (V, E)$?

9. An undirected graph $G = (V, E)$ is *bipartite* if V can be split into two sets L and R such that each edge has one vertex in L and the other vertex in R. Design an algorithm which decides whether or not G is bipartite in $O(|V| + |E|)$ time.

10. The reverse of a directed graph $G = (V, E)$ is another directed graph $G^R = (V, E^R)$ on the same vertex set, but with all edges reversed; that is, $E^R = \{(v, u) \mid (u, v) \in E\}$.

    (a) Give a $O(|V| + |E|)$ time algorithm for computing the reverse of a graph in adjacency lists format.

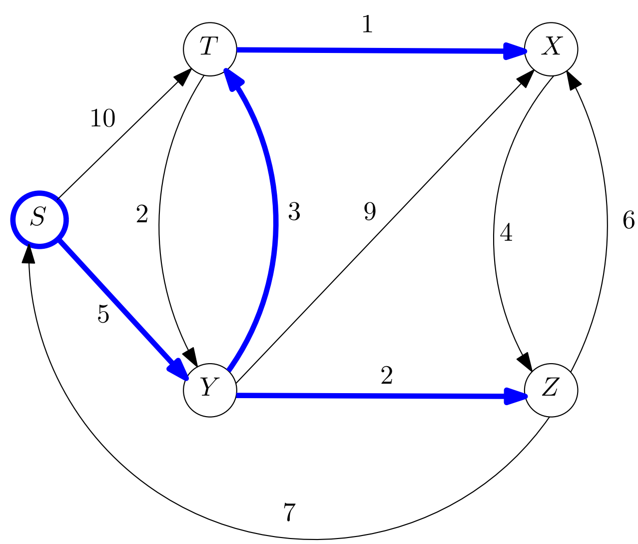
    (b) What time of computation do you get if you use the adjacency matrix format?

11. As you already know, if the weight of all edges in a graph is 1, we can use breadth-first search (BFS) to find the shortest path between two vertices. When the weights are not all equal, as you already know, we can use Dijskstra's algorithm.

    What do you think of the following approach as a shortest path algorithm when the weights of the edges are not all equal? Replace each edge with weight k by a path made of k edges, each of which has weight 1. Then, run BFS.

12. Can Dijkstra's algorithm be used to find shortest paths in a graph with some negative weights? Justify your answer.

13. In class, we did the trace of Dijkstra's algorithm on the following graph, starting at S.



    If we look carefully at the paths we obtain by running Dijkstra's algorithm, we get a tree rooted at S. This tree is called the *shortest-path tree rooted at S*. Explain how to modify Dijkstra's algorithm such that it returns the shortest path tree rooted at S. What is the running time of your algorithm?

—–Proofs—–

14. As we saw in class, a *tree* is an undirected graph that is connected and acyclic. Prove that a tree on n nodes has $n-1$ edges.

15. Explain how a vertex u of a directed graph can end up in a DFS-tree containing only u, even though u has both incoming and outgoing edges in $G = (V, E)$.

16. Let $G = (V, E)$ be an undirected graph.

    (a) Show that if G is acyclic, then $|E| \leq |V| - 1$.

    (b) Give an algorithm that decides if G contains a cycle in $O(|V|)$ time.

17. Prove or disprove. If a directed graph $G = (V, E)$ contains a path from u to v, then any depth-first search must result in $\text{pre}(v) < \text{post}(u)$.

18. Prove or disprove. If a directed graph $G = (V, E)$ contains a path from u to v, and if $\text{pre}(u) < \text{pre}(v)$ in a depth-first search of G, then v is a descendant of u in the DFS-forest produced.

19. Prove or disprove. In any connected undirected graph $G = (V, E)$, there is a vertex $v \in V$ whose removal leaves G connected.

20. Prove or disprove: Let $G = (V, E)$ be a directed graph such that for all vertices $v \in V$, we have $\text{indegree}(v) \geq 1$. Therefore, G has a cycle.

    Recall that indegree(v) is equal to the number of vertices $u \in V$ such that $(u, v) \in E$.

21. Let $G = (V, E)$ be a graph, where $V = \{s, x_1, x_2, ..., x_n, y_1, y_2, ..., y_n, t\}$ and the set of edges is defined as in Figure 1.

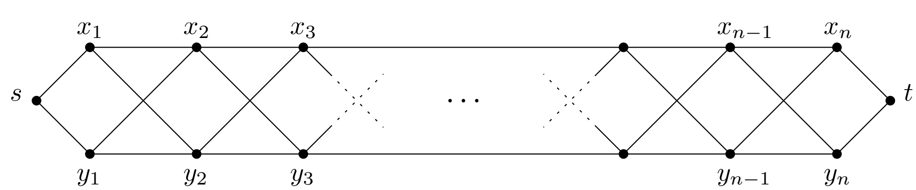


Figure 1:

    Suppose that the weight of all edges is 1. How many different shortest paths are there between s and t?

22. When is $O(n \log(n) + m)$ better than $O((m + n) \log(n))$?

23. Prove that any connected, undirected graph $G = (V, E)$ with $|E| = |V| - 1$ is a tree.

24. Prove that an undirected graph is a tree if and only if there is a unique path between any pair of nodes.