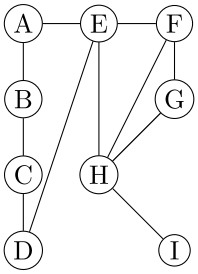
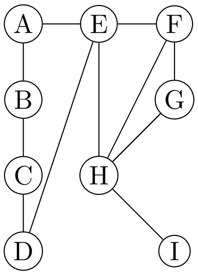
Problem 1

Run a Breadth First Traversal for graph G starting at A. Write the order the nodes are explored. Assuming breaking ties lexicographically (i.e., if more than one child pick the one that comes first in lexicographic order).



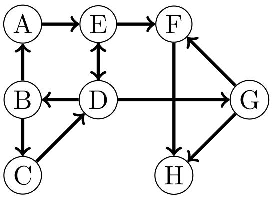
Problem 2

Run a Depth First Traversal for graph G starting at A. Write the order the nodes are explored. Assuming breaking ties lexicographically.



Problem 3

Consider the following directed graph G

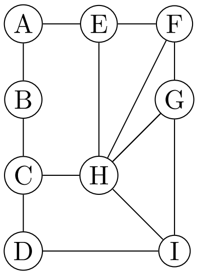


1. Give the adjacency matrix and the adjacency list representation of this graph
2. Is this graph strongly connected?
3. This graph has a few directed cycles. Find the minimum number of edges that you need to delete from this graph in order to make it a directed acyclic graph (DAG).

4. Write down a topological ordering of the resulting DAG.
5. Write down the transitive closure of the graph.

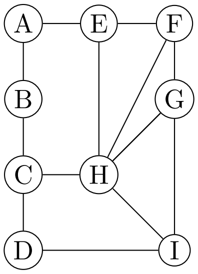
Problem 4

Run a Breadth First Search for graph G starting at A and searching for I. Write out the path that is explored. Assuming breaking ties lexicographically.



Problem 5

Run a Depth First Search for graph G starting at A and searching for I. Write out the path that is explored. Assuming breaking ties lexicographically.



Problem 6

An undirected graph $G = (V, E)$ is said to be bipartite if its vertex set V can be partition into two sets A and B such that $E \subseteq A \times B$. Design an $O(n + m)$ algorithm to test if a given input graph is bipartite using the following guide:

1. Suppose we run BFS from some vertex $s \in V$ and obtain layers $L_1, \ldots, L_k$. Let (u, v) be some edge in E. Show that if $u \in L_i$ and $v \in L_j$ then $|i - j| \leq 1$.
2. Suppose we run BFS on G. Show that if there is an edge (u, v) such that u and v belong to the same layer then the graph is not bipartite
3. Suppose G is connected and we run BFS. Show that if there are no intra-layer edges then the graph is bipartite
4. Put together all the above to design an $O(n + m)$ time algorithm for testing bipartiness.