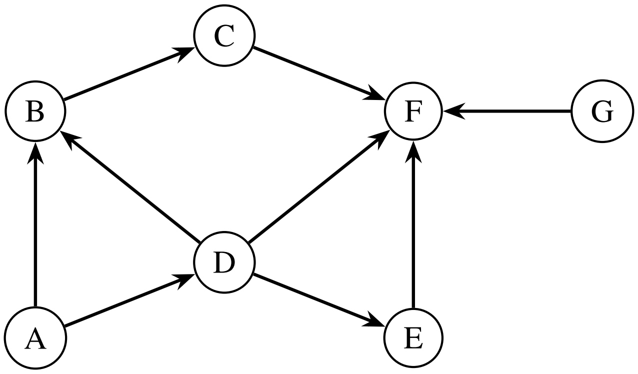
Discussion 11



1 Graph Representations

Write the graph above as an adjacency matrix, then as an adjacency list.

2 DFS and BFS

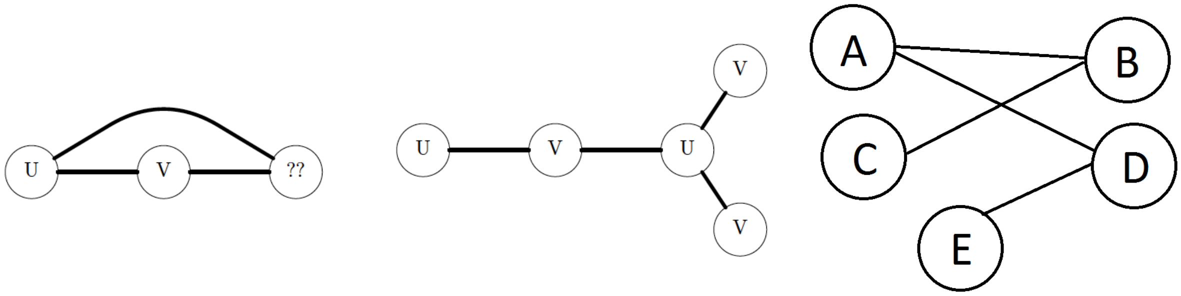
Give the DFS preorder, DFS postorder, and BFS order of the graph starting from vertex *A*. Break ties alphabetically.

3 Topological Sorting

Give a valid topological sort of the graph above. (Hint: Use the reverse postorder.)

4 Graph Algorithm Design: Bipartite Graphs

An undirected graph is said to be bipartite if all of its vertices can be divided into two disjoint sets U and V such that every edge connects an item in U to an item in V. For example, the graphs in the center and on the right are bipartite, whereas the graph on the left is not. Provide an algorithm which determines whether or not a graph is bipartite. What is the runtime of your algorithm?



5 Extra for Experts: Shortest Directed Cycles

Provide an algorithm that finds the shortest directed cycle in a graph in $O(EV)$ time and $O(E)$ space, assuming $E > V$.

6 Extra for Experts: DFS Gone Wrong

Consider the following implementation of DFS, which contains a crucial error:

```
create the fringe, which is an empty Stack
        push the start vertex onto the fringe and mark it
        while the fringe is not empty:
                pop a vertex off the fringe and visit it
                for each neighbor of the vertex:
                        if neighbor not marked:
                                push neighbor onto the fringe
                                mark neighbor
```

Give an example of a graph where this algorithm may not traverse in DFS order.