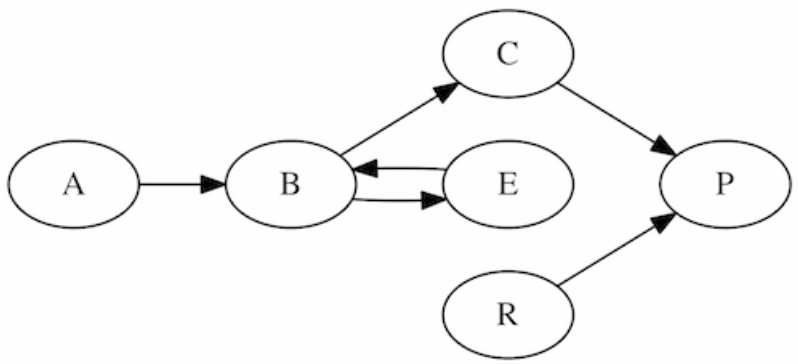


1 Graph Representations

For the graph above, draw the adjacency list and adjacency matrix representation.

2 DFS and BFS

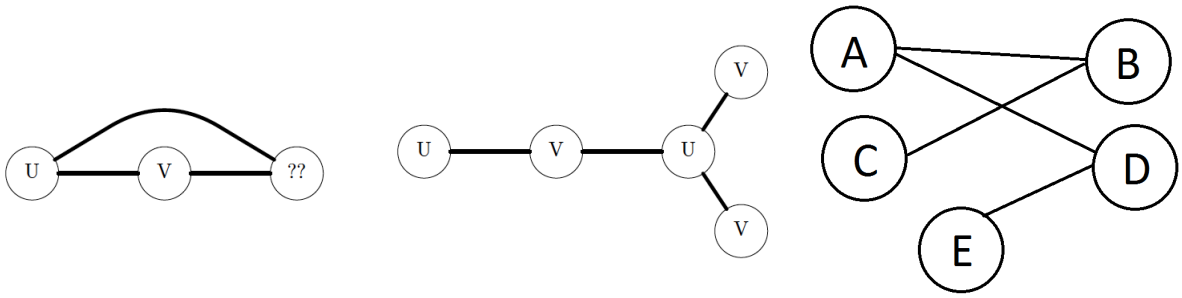
Give the DFS Preorder, DFS Postorder, and BFS order of the graph starting from vertex A. Whenever there is a choice of which node to visit next, visit nodes in alphabetical order.

3 Topological Sorting

Which edge would we need to remove so that there exists a topological sort for the graph above? Give a valid topological sort (Hint: Use DFS 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 Algorithm Design: 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: Daniel's Dare for the Daring

Master brogrammer, Edwin Edgehands decides to try his hand at implementing the Depth First traversal algorithm. Here is Edgehands' pseudocode:

```
Create a new Stack of Vertices
        Push the start vertex 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
```

Your TA, Joshua Shrug claims that the above traversal isn't quite DFS. Give an example graph where it may not traverse in DFS order.