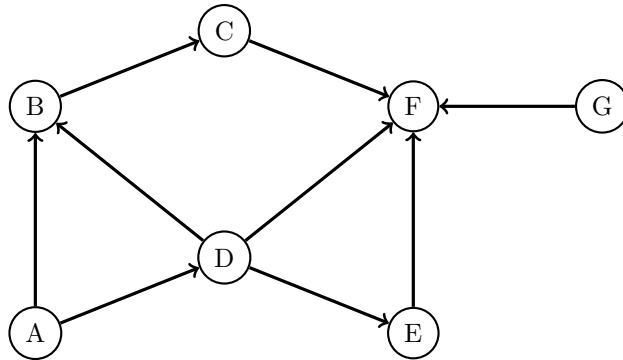


Graphs



- 1.1 Write the graph above as an adjacency matrix, then as an adjacency list. What would be different if the graph were undirected instead?

Skipped

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

pre : ABCFDE~~G~~
post : FCBEDA~~G~~

BFS: ABDCEF~~G~~

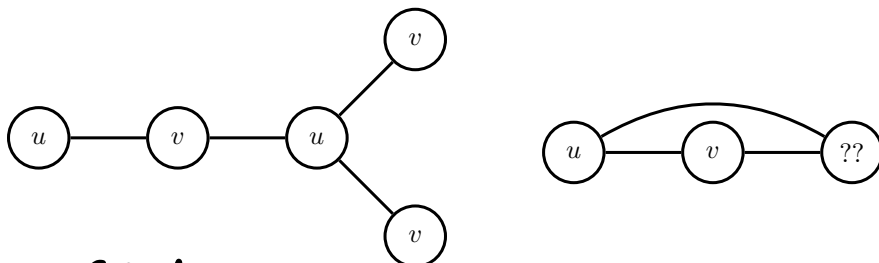
disconnect

- 1.3 Give a valid topological sort of the graph. (*Hint*: Consider the reverse postorder of the whole graph.)

G A D E B C F

Graph Algorithm Design

- 2.1 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 below, the graph on the left is bipartite, whereas on the graph on the right is not. Provide an algorithm which determines whether or not a graph is bipartite. What is the runtime of your algorithm?



DFS
will be
fine

Start marking from any vertices, until there are any contradictions. $\Rightarrow O(V+E)$

- 2.2 Provide an algorithm that finds the shortest cycle (in terms of the number of edges used) in a directed graph in $O(EV)$ time and $O(E)$ space, assuming $E > V$.

☆ Please note that in the problem, we assume we want to identify the shortest circle containing a particular source.

- 2.3** 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

```

ring and mark it
if not marked.

$$1 \rightarrow x \rightarrow \begin{matrix} 2 \\ 3 \\ 4 \end{matrix} \rightarrow \begin{matrix} 3 \\ 4 \end{matrix} \rightarrow \begin{matrix} 4 \\ 3 \\ 4 \end{matrix} \rightarrow \begin{matrix} 3 \\ 4 \end{matrix} \rightarrow 4$$

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



$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$.
we should mark only
when we actually visit a vertex