

## 1 Graph Representation

---

Represent the graph with edge list (a.k.a. adjacency list) and adjacency matrix representation.

## 2 Searches and Traversals

---

Run depth first search (DFS) and breadth first search (BFS) on the graph, starting from node A. List the order in which each node is traversed. Whenever there is a choice of which node to visit next, visit nodes in alphabetical order.

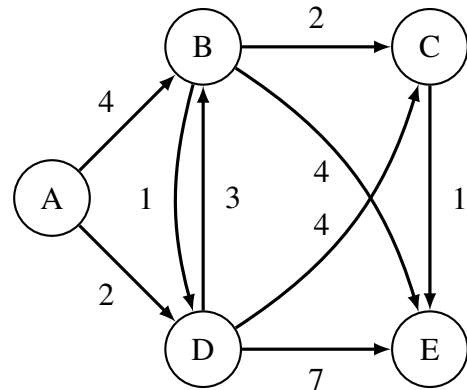
## 3 Topological Sorting

---

Give a valid topological ordering of the graph. Is it unique?

## 4 Dijkstra's Algorithm

Given the following graph, write down the value  $\text{dist}(v)$  for all vertices  $v$  during each iteration of Dijkstra's algorithm, starting at node A.



## 5 Exercise: Bipartite Graphs

An undirected graph is a bipartite graph if its vertices can be separated into two disjoint sets such that each edge in the graph spans both sets (is connected to a vertex in each set). Given a connected graph  $G$ , fill in the method below so that it returns `True` iff  $G$  is a bipartite graph.

```
public static boolean isBipartite(Graph G) {
    Node start = getRandomNode(G);
    HashMap<Node, Boolean> visited = new HashMap<Node, Boolean>();
    ArrayList<Node> fringe = new ArrayList<Node>();

    _____;

    _____;

    while ( _____ ) {
        _____;

        _____;

        for ( _____ ) {
            if ( _____ )
                _____;
            else {
                _____;

                _____;

                _____;
            }
        }
    }

    _____;
}
```