Recall that given a graph \( G = (V, E) \), we write \( n = |V| \) for the number of vertices and \( m = |E| \) for the number of edges.

1. (2 points) Consider an undirected graph \( G = (V, E) \) with \( n = 5 \) given in the following adjacency list representation:

   \[
   \begin{aligned}
   \text{Adj}[1] &= [2, 3] \\
   \text{Adj}[2] &= [1, 4, 5] \\
   \text{Adj}[3] &= [1, 4] \\
   \text{Adj}[4] &= [2, 3] \\
   \text{Adj}[5] &= [2] \\
   \end{aligned}
   \]

   (a) Draw the graph \( G \). Write down the adjacency matrix representation of \( G \).

   (b) Draw the BFS and DFS trees of \( G \) starting from node 1. (Assume that when exploring a node, the algorithm iterates over its neighbors from smaller index to larger index.)

2. (2 points) Consider an undirected graph \( G \) given in the adjacency list representation. Design an \( O(m + n) \) algorithm to decide whether \( G \) has a (simple) cycle or not. Prove the correctness of your algorithm and analyze its runtime.

   (You can use the properties and the runtime of BFS or DFS that we stated in class without proving them. Your algorithm does not need to output the cycle.)

3. (2 points) List all the topological orderings of the following directed (acyclic) graph.

   1
   \[ \rightarrow \]
   2
   \[ \rightarrow \]
   4
   \[ \rightarrow \]
   6
   \[ \rightarrow \]
   3
   \[ \rightarrow \]
   5

4. (extra credit, 1 point) Let \( G = (V, E) \) be an undirected connected graph. Fix a vertex \( s \in V \). Suppose we run BFS and DFS on \( G \) starting at node \( s \). Let \( T_B \) and \( T_D \) denote the resulting BFS tree and DFS tree respectively. Prove that if \( T_B = T_D \), then \( G \) must be a tree.
(Possible Hint: Let $T = T_B = T_D$. We know that $G$ contains all edges of $T$ so we only need to prove that $T$ contains all edges of $G$. Consider an edge $e = (u, v)$ of $G$, we can prove $e$ must appear in $T$. More specifically, we can show that one of $u$ and $v$ must be an ancestor of the other, and $u$ and $v$ must appear on the same or adjacent levels.)