1. There are two general approaches for traversing a graph from some starting vertex $s$:
   - Depth First Search (DFS) where you explore as deeply into the graph as possible. If you reach a “dead end,” we backtrack to the deepest vertex that allows us to try a different path.
   - Breadth First Search (BFS) where you find all vertices a distance 1 (directly connected) from $s$, before finding all vertices a distance 2 from $s$, etc.

What data structure would be helpful in each type of search? Why?

   a) Depth First Search (DFS):

   b) Breadth First Search (BFS):

2. How could you use the bfs algorithm to solve the word ladder problem?

3. How would you augment the LinkedVertex and LinkedDirectedGraph classes to keep track of the solution to the word ladder?