This quiz has 3 questions, for a total of 10 points.

1. 4 points Fill in the blanks to complete this implementation of a adjacency list representation of a directed graph.

   ```python
   class Edge:
       def __init__(self, src, tgt):
           self.source = src
           self.target = tgt
   class DirectedAdjList:
       def __init__(self, num_vertices):
           self.array = [___(a)___ for i in range(0,num_vertices)]
       # add edge (u,v) to the graph
       def add_edge(self, u, v):
           self.array[u].append(v)
       # return the list of edges whose source is vertex u
       def out_edges(self, u):
           return [___(b)___ for v in ___(c)___]
       # return an Edge or None
       def find_edge(self, u, v):
           for w in self.array[u]:
               if ___(d)___:
                   return Edge(u,v)
           return None
   ```

   Solution:
   
   (a) []
   (b) Edge(u,v)
   (c) self.array[u]
   (d) w == v

2. 3 points Identify a breadth-first tree rooted at node g in the following directed graph by drawing dark lines over the edges in the breadth-first tree.

   ![Graph Diagram]

   Solution: One of the possible solutions:

   ![Solution Diagram]
3. [3 points] Identify a depth-first tree rooted at node $g$ in the following directed graph by drawing dark lines over the edges in the depth-first tree.

Solution: One of the possible solutions: