Data Structures and Algorithms II
Fall 2021 Section 2
Midterm

Name: ____

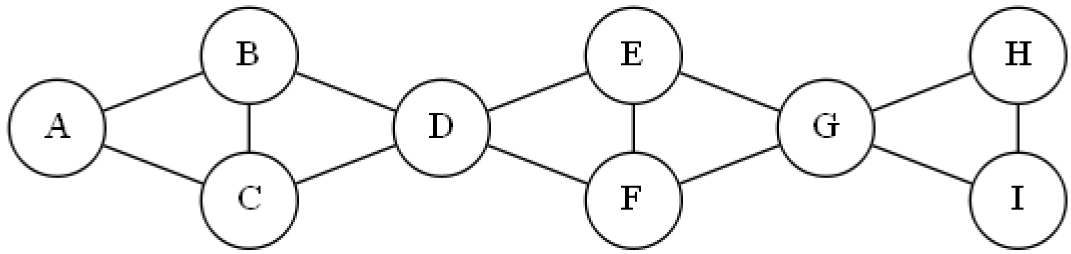
E-mail: ____

Write all of your answers directly within this test booklet. Use the backs of pages if necessary. This is an open-book, open-notes test. You should not use any electronic devices. You may use whatever scrap paper you like, but do not hand it in.

Please also write your name at the top of each page.

(1)

(1) Consider the following undirected graph:



(a) In order to determine if this graph is biconnected, you apply the algorithm discussed in class. Draw the depth-first spanning tree that results if you start with D as the root. Whenever you have a choice of which node to visit next, choose the node that comes first alphabetically (i.e., choose the letter that comes first in the alphabet). Indicate tree edges as solid arrows, and back edges as dashed arrows.

(b) Next to each node in the depth-first spanning tree that you drew for part (a), indicate the Num and Low parameters of each node. (To the right of each node in your drawing, specify the Num value, followed by a slash, followed by the Low value.)

(c) Which nodes are articulation points? For each node that you list, explain how you can determine that the node is an articulation point using Num and Low values of nodes.

(d) Is the graph biconnected? How do you know?

(2) In class, we have examined the following pseudo-code for Prim's algorithm:

```
MinimumSpanningTreePrim (Graph G, Vertex s)
   for each vertex v in G
     dᵥ ← ∞
     knownᵥ ← FALSE
   dₛ ← 0
   pₛ ← NULL
   while there are still unknown vertices
     v ← the unknown vertex with the
         smallest d-value
     knownᵥ ← TRUE
     for each edge from v to vertex w
       if c_v,w < d_w
         d_w ← c_v,w
         p_w ← v
```

Briefly answer the following questions related to this algorithm:

(a) The algorithm assumes that the graph is connected. Assume that you want the algorithm to detect when the graph is not connected. Explain how you can modify that pseudo code above to detect when a graph is not connected, and to return FALSE when this is detected. If the graph is connected, the algorithm should return TRUE at the end. Indicate what should be added to the pseudo-code below to the side of the pseudo-code above, with arrows pointing to the locations where any new code should be inserted.

(b) Suppose that the algorithm has already been run for a connected graph. How can you use the d-values and/or p-values that have been determined to calculate the total cost of the minimum spanning tree?

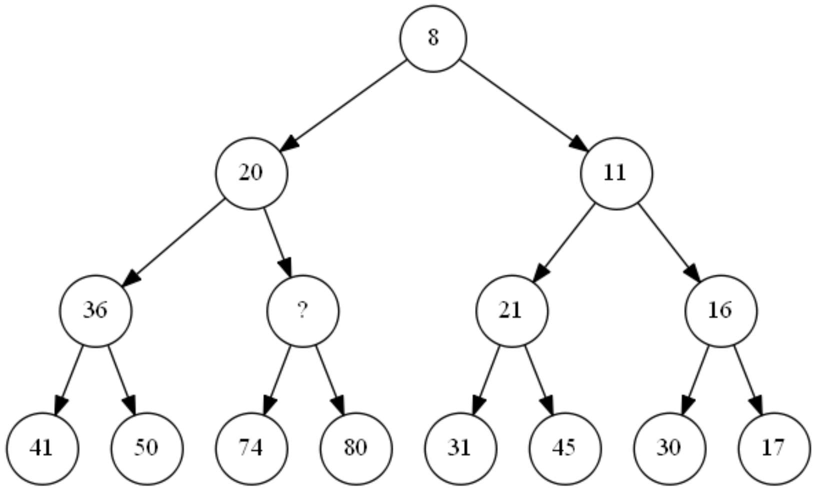
(c) Suppose that the algorithm has already been run for a connected graph. Below, write simple pseudo-code that uses the d-values and/or p-values that have been determined to display all the edges that are part of the minimum spanning tree. That is, your goal is to display a set of pairs (v, w), where each (v, w) is an edge in the minimum spanning tree. The order that you display the edges does not matter. (You don't have to specify this as a function; just assume you have access to the graph, G, and to all of the d-values and p-values.)

(3) Answer the following questions related to binomial queues:

(a) Draw all possible binomial queues with exactly four total nodes, containing the keys 1, 2, 3, and 4. (Indicate the key value inside each node.)

(b) How many ways can you create a binomial queue with five total nodes, containing the keys 1, 2, 3, 4, and 5. I am NOT asking you to draw them. Just indicate how many there are. Briefly explain how you figured out the answer. HINT: Your answer to part (a) might be helpful to consider.

(4) Consider the following binary heap:



The values drawn inside the nodes are the keys of the nodes. Note that one of the keys, marked with a "?", is unknown. Call that node X.

(a) Assuming that all keys are integers, not necessarily distinct, what is the full range of values that the key of X might have?

(b) In class, we mentioned two ways to delete a node. One way is to decrease the node's key to the root's key minus one, allowing it to percolate appropriately, and then to perform a deleteMin operation. Use this method to delete node X. Draw the resulting heap below.

(c) The second method of deleting a node is to move the final node in the heap to the current position, and then allowing it to percolate appropriately. Consider using this method to delete node X. Would the resulting heap in this case be the same or different than the answer to part (b)? If it is different, draw the resulting heap below.

(d) Starting with the original heap as shown on the previous page (with the key of X unknown), what is the least number of insertions into that heap that might cause X to move via a percolation?

(5) Briefly answer the following questions related to graphs and graph algorithms.

(a) When adjacency lists are used to represent a directed graph, in how many lists does each edge appear?

(b) When adjacency lists are used to represent an undirected graph, in how many lists does each edge appear?

(c) If Dijkstra's algorithm, as discussed in class, is applied to a graph containing a negative cycle, might it get stuck in an infinite loop? Briefly explain your answer.

(d) If the Bellman-Ford algorithm is applied to a graph that has no negative-cost edges, is it guaranteed to produce optimal results? Briefly explain your answer.

(e) If the Ford-Fulkerson method, as discussed in class, is applied to a flow network in which t is not reachable from s (i.e., there is no path from s to t in the graph), what would happen?