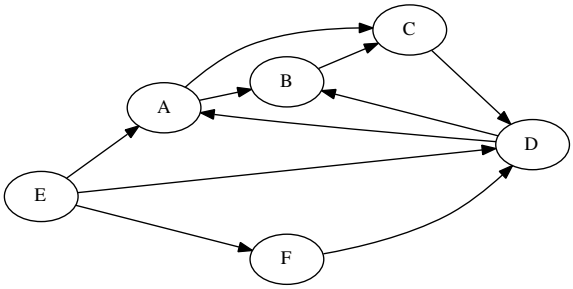
Name: _____

1. (1 pt.)
   - **Read all material carefully.**
   - You may refer to your books, papers, and notes during this test.
   - No computer or network access of any kind is allowed (or needed).
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use textbook and classroom conventions for notation, algorithmic options, etc.
   - Ask for clarifications on the above if needed.

   Write your name in the space provided above.



2. (29 pts. for all parts) Answer the following briefly for the directed graph G depicted above, using definitions discussed in class.

   (a) (1 pt.) What is the graph's *order*?

   (b) (1 pt.) What is the its *size*?

   (c) (4 pts.) How many *directed simple cycles* does it have? Write down each such cycle below by listing its vertices, in order.

   (d) (4 pts.) Temporarily treating all edges as undirected, how many *undirected simple cycles* does it have? Write down each such cycle below by listing its vertices, in order.

   (e) (4 pts.) How many *strongly connected components* does it have? List the vertices in each strongly connected component.

(f) (5 pts.) Depict **all** *subgraphs* of G that have vertex set $\{A, B, C\}$.

(g) (5 pts.) Using notation used in class and the textbook, depict the *adjacency list* representation of G.

(h) (5 pts.) Depict the *adjacency matrix* of G.

3. (15 pts.) Recall the definition of *k-sorted* sequences from the discussion of shellsort. For each of the following, either provide a requested sequence or prove that no such sequence exists. **Explain your answers briefly.**

   (a) A sequence that is 3-sorted but not 7-sorted.

   (b) A sequence that is 7-sorted but not 3-sorted.

   (c) A sequence that is 2-sorted but not 4-sorted.

   (d) A sequence that is 4-sorted but not 2-sorted.

[additional space for answering the earlier question]

4. (15 pts.) Using *merge-based insertions* and *skew-merging*, trace the state of an initially empty *skew heap* as a result of the following keys being inserted in order. Depict the state of the heap after each insertion.

$$66, 70, 99, 43, 58, 77, 41, 74, 81, 70$$

Apply skew merging carefully. Depict intermediate working trees and use them to **check your work as you proceed** in order to avoid propagating errors.

[additional space for answering the earlier question]