Name: ______________________________________

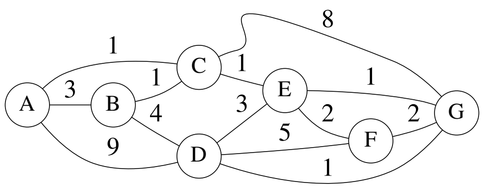
1. (1 pt.)
   - **Read all material carefully.**
   - *If in doubt whether something is allowed, ask, don't assume.*
   - You may refer to your books, papers, and notes during this test.
   - E-books may be used *subject to the restrictions* noted in class.
   - No computer or network access of any kind is allowed (or needed).
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use class and textbook conventions for notation, algorithmic options, etc.
   - Budget your time: roughly one minute per point.

   Write your name in the space provided above.

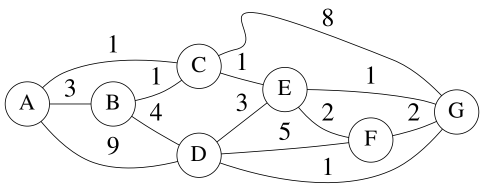
WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

2. (19 pts.) Trace the action of Dijkstra's single-source shortest-path algorithm on the following graph with source vertex A, using the textbook's Fig. 24.6 (p.659) as a guide. In particular:

   - Depict the state of the algorithm after each iteration of the while loop in the textbook's pseudocode.
   - Highlight edges that determine predecessor values using double-lines.
   - Depict the state of the priority queue Q (include objects and keys).

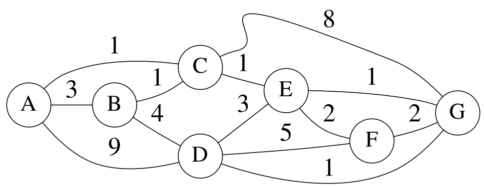


[additional space for answering the earlier question]

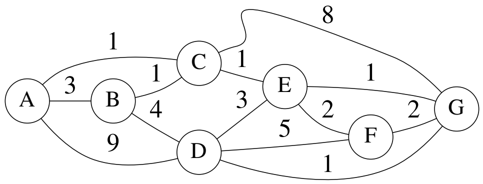


3. (20 pts.) Trace the operation of MST-PRIM on the graph of Question 2 (also below) from starting vertex A using the conventions of Figure 23.5 (p. 635) of the textbook, **but augmented** with depictions of the priority queue used by the algorithm. In particular:

   - Depict the state of the algorithm after each iteration of the while loop in the textbook's pseudocode.
   - Highlight edges belong to partial spanning tree A using double-lines.
   - Depict the state of the priority queue Q (include objects and keys).



[additional space for answering the earlier question]



4. (20 pts.) Use the textbook's method of reducing 3-CNF-SAT to SUBSET-SUM to map the following instance of 3-CNF-SAT to the appropriate instance of SUBSET-SUM.

$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee x_4)$

[additional space for answering the earlier question]

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee x_4)$$

5. (20 pts.) Answer each part clearly.

   (a) Define the term *derangements* as used in class.

   (b) List all derangements of the sequence $(1, 2, 3, 4)$.

   (c) Provide pseudocode for an algorithm for computing the *number* of derangements of a sequence of length n (not the derangements themselves).

   (d) Explain clearly why your algorithm is correct

   (e) State the running time of your algorithm as a function of n.

   (f) Justify your running-time claim.

   (g) Is the algorithm's running time polynomial in its input size? Explain briefly.

[additional space for answering the earlier question]

6. (20 pts.) Solve the recurrence

$$T(n) = 8 \cdot T(n/9) + 43 \cdot n \cdot \log\log n + 42$$

to determine a function f such that

$$T(n) = \Theta(f(n))$$

Clearly state the method you use and outline its key steps. (Show your work.)