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 and electronic resources may be used, *but only as a library* without direct communication with sentient beings.
   - No interactive communication (other than that required for class) is permitted.
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use class and textbook conventions for notation, algorithmic options, etc.

   Write your name in the space provided above.

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

Do not write in the following table.

Q	Full	Score
1	1	
2	29	
3	30	
4	20	
5	20	
total	100	

2. (29 pts.) Using the notational conventions used for the detailed worked example of heapsort discussed in class, trace the action of the textbook's implementation of heapsort on an array with the following elements, in order:

   `29 57 39 23 54 67 86 94 85 18 74 19`

   Be sure to depict the tree and array representations at each step as described in class, as well as to highlight the changes in the tree resulting from the percolation operations. Do not forget to do the same for the initial build-heap stage as well.

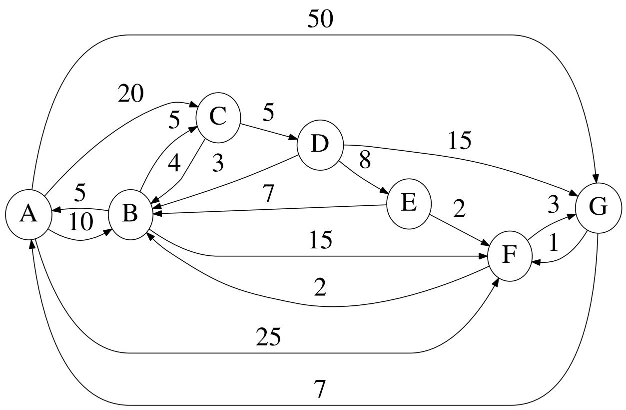
   [Note the points/time budget for this question; do not skip details. Use extra pages if needed.]

[additional space for answering the earlier question]

[additional space for answering the earlier question]

[additional space for answering the earlier question]

3. (30 pts.) Refer to Figure 15.26 (p. 512) in the textbook, which depicts a trace of Dijkstra's algorithm on the sample graph with V_0 as the source vertex. Depict the *first six iterations* (after initialization) of a similar trace on the *following graph with A as the source vertex*. Ensure that you *clearly* indicate *all* the details following the conventions of that figure precisely.



[additional space for answering the earlier question]

[additional space for answering the earlier question]

[additional space for answering the earlier question]

[additional space for answering the earlier question]

4. (20 pts.) Using precisely the format and conventions of Figure 15.5 (p. 495) of the textbook, depict the data structure corresponding to the final graph in your answer to Question 3.

5. (20 pts.) Provide standard C++ code, as detailed as you can, to perform the following actions in sequence:
   (a) Create a STL *map* data structure that maps the names (strings) of US states to their populations (integers). Use case-insensitive string matching for the names of states.
   (b) Insert data for at least three states. (You can make up the data; it needn't be real.)
   (c) Create a STL priority queue (a max heap) for the state-population records, with the key being the populations.
   (d) Iterate over the records in the above map in alphabetical (lexicographical) order of state names, inserting the name and population of each state thus visited into the priority queue.
   (e) Dequeue and print the names and populations of the most populous three states by using the priority queue's primary operations.
   (f) Ensure that any memory used by the above data structures is freed up; then print "all done; next?" on standard output and wait for user input on standard input. (No code for further actions needed.)

   Your code and comments must make correctness very easy to discern.

[additional space for answering the earlier question]

[additional space for answering the earlier question]