**Name:** _____

Please submit this homework by following the homework submission instructions on the class Web site. Reminder: You are welcome, and encouraged, to use any resources (e.g., Web sites) to help you with your work. However, *all such help must be clearly noted* in your submissions. Further, no matter what you use, *you must be able to explain* how and why it works. Refer to the class policy for details, and ask for clarifications if you are unsure if something is allowed. Questions marked with ⋆ are optional and may be answered for extra credit.

1. (1 pt.) Write your name in the space provided above.

2. (1 pt.) Package and submit your solutions to the programming questions as in previous assignments. After uploading your jar file to the FTP server, complete the following: File name: _____ Size, in bytes: _____

3. Read the textbook's description of graphs and their representation[1] and answer the following:

   (a) (5 pts.) For each statement below, indicate whether it is true or false by writing *True* or *False* in the blank next to it. *Justify your answer briefly.*

      i. The vertices in a simple path are all distinct. _____

      ii. The length of the path equals the number of vertices on the path. _____

[1] Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 3rd edition (Addison-Wesley, 2006), Section 11.1, pp. 471–483.

iii. The space required by the adjacency matrix representation of a graph with $n$ vertices and $e$ edges is $\Theta(n + e)$. _____

iv. The space required by the adjacency list representation of a graph with $n$ vertices and $e$ edges is $\Theta(n^2 + e)$. _____

(b) (5 pts.) Assuming the adjacency-list implementation of graphs described in the textbook, provide Java code for a method `Graph.renameVertex(String oldName, String newName)` that changes the name of the vertex named `oldName` to `newName`, *updating all the data structures as needed to maintain a consistent representation of the graph.*

(c) (5 pts.) Describe, as precisely as possible, the effect of replacing the definition of `INFINITY` in the `Graph` class[2] with the following:

```
public static final double INFINITY = Double.POSITIVE_INFINITY;
```

(d) (5 pts.) Explain how the change described by the footnote on page 478 may be implemented. What additional data structures are needed? How does the time required to access the affected information change?

[2] *Idem*, line 20 of Figure 14.8, p. 480.

(e) (5 pts.) Explain, as precisely as possible, the format of the input file that is used by the `main` method of `Graph` to read a graph. Provide a brief but representative example of the contents of such a file, along with the resulting graph.

(f) (5 pts.) How are the `prev` and `dist` members of the `Vertex` class used?

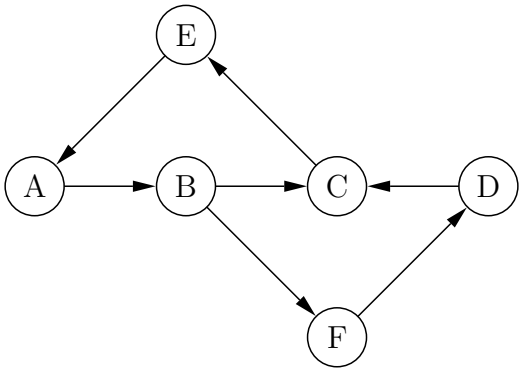
4. Read the textbook's description of the unweighted shortest-path problem and its implementation[3] and answer the following:

   (a) (4 pts.) For each statement below, indicate whether it is true or false by writing *True* or *False* in the blank next to it. *Justify your answer briefly.*

      i. There is always a unique shortest path between any pair of vertices in a graph. _____

      ii. The unweighted shortest-path algorithm visits vertices of a graph in breadth-first order _____

   (b) (10 pts.) Illustrate the operation of the unweighted shortest-path algorithm on the following graph, with A as the start vertex. Use a scheme similar to that of Figure 14.21 in the textbook. However, instead of shading visited nodes, mark them with a *; similarly, instead of the light shading for the eyeball's position, mark the position using a +. Be sure to label the vertices with their distances as well, as done in the textbook.




[3] *Idem*, Section 14.2, pp. 483–489.

(c) (4 pts.) Explain, as precisely as possible, the effect of replacing line 27 of Figure 14.22 with the following:

```
q.addFirst(w);
```

Does the resulting implementation produce correct results? Are there any changes in the output? Is the running time different?

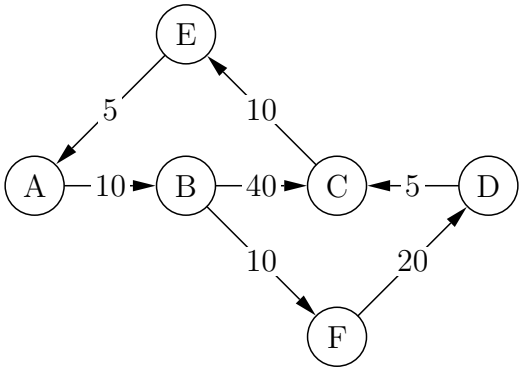
5. Read the textbook's description of the positive-weighted shortest-path problem and its implementation[4] and answer the following:

   (a) (4 pts.) For each statement below, indicate whether it is true or false by writing *True* or *False* in the blank next to it. *Justify your answer briefly.*

      i. Unlike the unweighted shortest-path algorithm, the weighted one may need to update the value of $D_w$ multiple times per vertex $w$. _____

      ii. By using a pairing heap implementation for priority queues, and the *decreaseKey* operation, we can reduce the asymptotic running time of method `dijkstra` of Figure 14.27. _____

   (b) (10 pts.) As in Question 4b, illustrate the operation of the positive-weighted shortest-path algorithm on the following graph, with A as the start vertex.




[4] *Idem*, Section 14.3, pp. 489–496.

(c) (6 pts.) Explain, as precisely as possible, the effect of replacing line 34 of Figure 14.27 with the following:

```
if( w.dist >= v.dist + cvw )
```

Does the resulting implementation produce correct results? Are there any changes in the output? Is the running time different?

6. For any positive integer $n$, consider the set $V$ composed of all subsets of the set $[n] = \{1, 2, \ldots, n\}$. We define a graph $B_n = (V, E)$ where $E = \{(u, v) \in V \times V \mid v \subsetneq u \ \wedge \nexists w \in V : v \subsetneq w \subsetneq u\}$. Intuitively, an edge $(u, v)$ in this graph indicates that $v$ is a maximal proper subset of $u$ in the sense that there is no proper subset $w \subsetneq u$ that is a proper superset of $v$.

   (a) (5 pts.) Draw the graphs $B_1$, $B_2$, $B_3$, and $B_4$ following the usual conventions. Label each node by the subset it represents. Represent an edge $(u, v)$ by an arrow from $u$ to $v$. Leave enough empty space around each vertex because additional labels will be added to each vertex by some questions below.

(b) (5 pts.) How many vertices does $B_n$ contain (as a function of $n$)? Justify your answer.

(c) (10 pts.) $\star$ How many edges does $B_n$ contain (as a function of $n$)? Justify your answer.

(d) (5 pts.) Does $B_n$ contain any directed cycles? Why?

(e) (5 pts.) Does $B_n$ contain any undirected cycles? Why?

(f) (5 pts.) What is the number of distinct functions $f : V \to [m]$ (as a function of $n$ and $m$)? (Recall our notation $[m] = \{1, 2, \ldots, m\}$.)

(g) (5 pts.) Given a graph $G = (V, E)$ and a subset $U \subseteq V$ of its vertices, we define the *subgraph induced* by $U$ as the graph $G_U = (U, E \cap (U \times U))$. Intuitively, $G_U$ consists of the vertices in $U$ and those edges of $G$ that are incident on vertices in $U$. How many induced subgraphs does $B_n$ have (as a function of $n$)? Why?

(h) (10 pts.) $\star$ Consider the set $S(n, m)$ of pairs $(G_n, f_m)$ where $G_n$ is an induced subgraph of $B_n$ (as in Question 6g) and $f_m$ is a function mapping vertices of $G_n$ to $[m]$ (as in Question 6f). How many elements does $S(n, m)$ contain (as a function of $n$ and $m$)? Why?