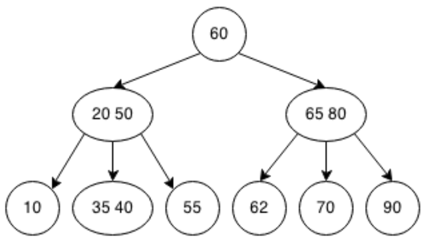
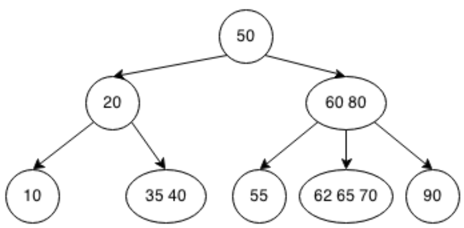
Homework 11: Graphs

1 Chapter 29: Balanced Search Trees

3. Add 62 and 65 to the 2-3 tree in Figure 29-27b
   - Show the final resulting tree



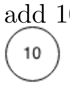
4. Add 62 and 65 to the 2-4 tree in Figure 29-27c
   - Show the final resulting tree



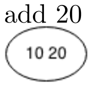
8. What tree results when you add the values 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 to each of the following initially empty tree?
   - Show what the tree looks like after **each** element is inserted. You do not need to show the "splitting" steps, but show what the tree looks like after each insertion. (There should be 10 trees shown for 8b and 10 for 8c).

   (b) A 2-3 tree

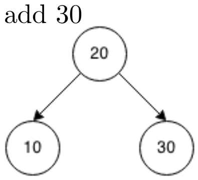
      i. add 10



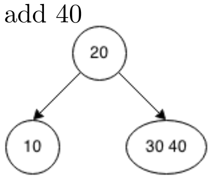
      ii. add 20



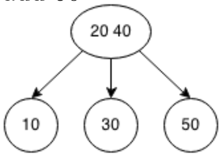
      iii. add 30



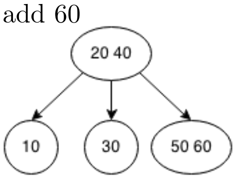
      iv. add 40



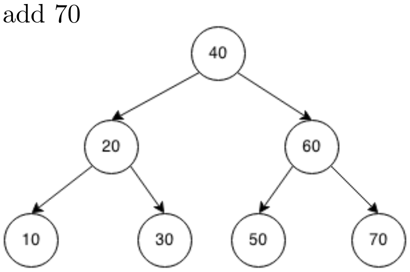
v. add 50



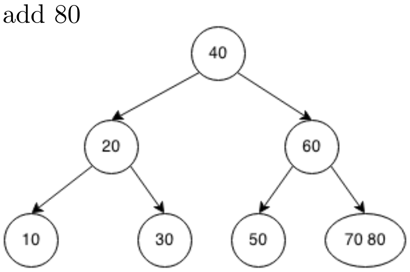
vi. add 60



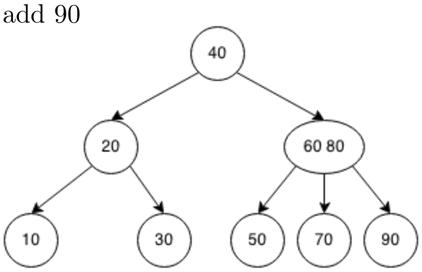
vii. add 70



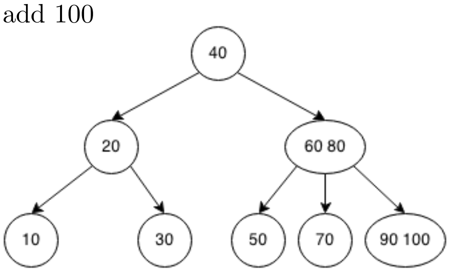
viii. add 80



ix. add 90

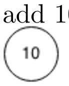


x. add 100

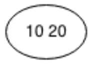


(c) A 2-4 tree

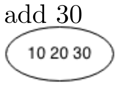
i. add 10



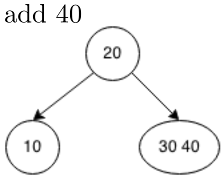
ii. add 20



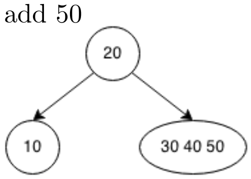
iii. add 30



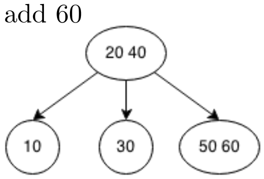
iv. add 40



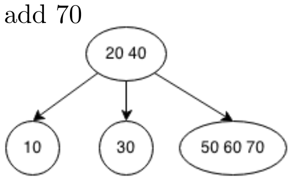
v. add 50



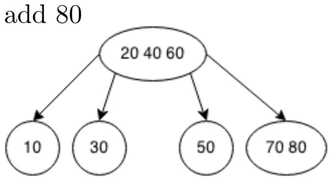
vi. add 60



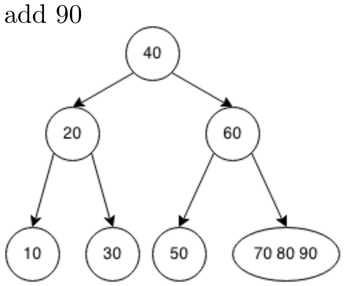
vii. add 70



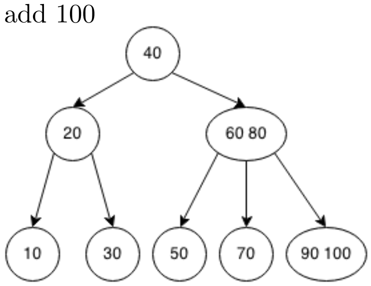
viii. add 80



ix. add 90



x. add 100



2 Chapter 30: Graphs

2. Describe each graph in Figure 30-22, using the terms introduced in Segments 30.1 through 30.4.

   - Describe the following for each graph:
     - directed/undirected
     - number of nodes
     - number of edges
     - unweighted/weighted
     - connected/disconnected
     - complete/not complete
     - cyclic/acyclic

   (a)
   - undirected
   - 5 vertices
   - 4 edges
   - unweighted
   - disconnected
   - not complete
   - cyclic

   (b)
   - undirected
   - 5 vertices
   - 8 edges
   - unweighted
   - connected
   - not complete
   - cyclic

   (c)
   - undirected
   - 4 vertices
   - 5 edges
   - unweighted
   - connected
   - not complete
   - cyclic

4. In what order does a breadth-first traversal visit the vertices in the graph in Figure 30-10 when you begin at

   (a) Vertex G

   My choice for traversal order was based on alphabetical order.
   GHIFCBE

   (b) Vertex F

   My choice for traversal order was based on alphabetical order.
   FCHBIE

5. Repeat the previous exercise, but perform a depth-first traversal instead.
   (a) Vertex G

   My choice for traversal order was based on alphabetical order.
   GHIFCBE

   (b) Vertex F

   My choice for traversal order was based on alphabetical order.
   FCBEHI

6. Consider the directed graph that appears in Figure 30-10, and remove the edge between vertices E and F, and the edge between vertices F and H.
   (a) In what order will a breadth-first traversal visit the vertices when you begin at vertex A?

   My choice for traversal order was based on alphabetical order.
   ABDEGHIFC

   (b) Repeat part *a*, but perform a depth-first traversal instead.

   My choice for traversal order was based on alphabetical order.
   ABEHIFCDG

8. Construct the topological ordering for the weighted, directed, acyclic graph in Figure 30-23.

   AFBCDEMLJKHGI

11. A tree is a connected graph without cycles.
   (a) What is the smallest number of edges that could be removed from the graph in Figure 30-1 to make it a tree?

   Only 1 edge needs to be removed.

   (b) Give one example of such a set of edges.

   One example is to remove the edge connecting Orleans and Chatham.

16. Find a map of the routes of a major U.S. airline. Such maps are usually printed at the back of in-flight magazines. You could also search the Internet for one. The map is a graph like the one in Figure 30-6. Consider the following pairs of cities:
   - Providence (RI) and San Diego (CA)
   - Albany (NY) and Phoenix (AZ)
   - Boston (MA) and Baltimore (MD)
   - Dallas (TX) and Detroit (MI)

- Charlotte (NC) and Chicago (IL)
- Portland (ME) and Portland (OR)

- I recommend using the interactive map on the Jet Blue website: http://www.jetblue.com/WhereWeJet/. If you use this, use Buffalo, NY instead of Albany, NY. If you use a different airline, you must include the map or the URL in your answer.

(a) Which pairs of cities in this list have edges (nonstop flights) between them?
  - Boston (MA) and Baltimore (MD)

(b) Which pairs are not connected by any path?
  - Providence (RI) and San Diego (CA)

(c) For each of the remaining pairs, find the path with the fewest edges.
  - Buffalo (NY) and Phoenix (AZ) have 2 edges
    - Buffalo (NY) and Boston (MA)
    - Boston (MA) AND Phoenix (AZ)

    - Buffalo (NY) and New York City (NY)
    - New York City (NY) and Phoenix (AZ)
  - Dallas (TX) and Detroit (MI) have 2 edges
    - Dalls (TX) and Boston (MA)
    - Boston (MA) and Detroit (MI)
  - Charlotte (NC) and Chicago (IL) have 2 edges
    - Charlotte (NC) and New York City (NY)
    - New York City (NY) and Chicago (IL)
  - Portland (ME) and Portland (OR) have 2 edges
    - Portland (ME) and New York City (NY)
    - New York City (NY) and Portland (OR)

EC. Describe real-world situations that could be represented by a graph with the properties below. For each graph, describe what the nodes are and what an edge represents.

- directed and disconnected

  Offered routes of airlines can be directed and disconnected. An offered path from one city to another may be different when the source and destination vertices are reversed, so the graph is directed. There may be offered paths to fly in a small area in the world that do not have an offered path to anywhere else in the world, so the graph will be disconnected.

- undirected and disconnected

  Friends lists on Facebook all over the world. Each friend that a person has on Facebook must be mutual between the two parties, so the edges are undirected. There may be a group of friends that are only friends with each other, and nobody else. Thus the graph is disconnected.

- complete

  The social relationship between a happy household. There are households where everyone in the household love each other, including themselves. This would create a complete graph.

3 Chapter 31: Graph Implementations

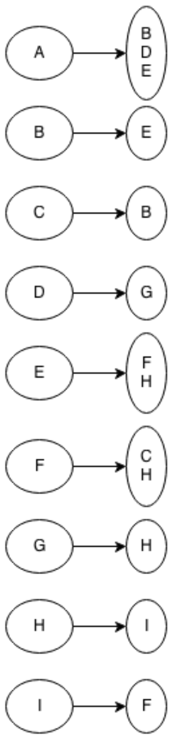
1. What adjacency matrix represents the graph in Figure 30-15a of the previous chapter?

	A	B	C	D	E	F	G	H	I
A		T		T	T				
B					T				
C		T							
D							T		
E						T		T	
F			T					T	
G								T	
H									T
I						T			

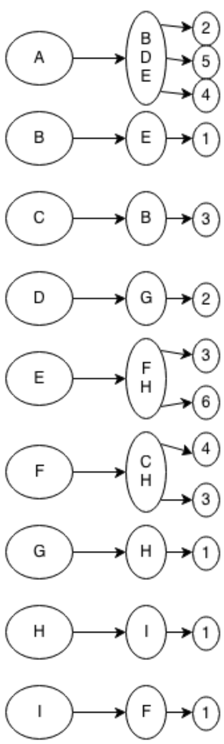
2. What adjacency matrix represents the graph in Figure 30-18a of the previous chapter?

	A	B	C	D	E	F	G	H	I
A		2		5	4				
B					1				
C		3							
D							2		
E						3		6	
F			4					3	
G								1	
H									1
I						1			

3. What adjacency lists represent the graph in Figure 30-15a of the previous chapter?



4. What adjacency lists represent the graph in Figure 30-18a of the previous chapter?



6. Suppose that you want only to test whether an edge exists between two particular vertices. Does an adjacency matrix or an adjacency list provide a more efficient way of doing this?

An adjacency matrix is more efficient because it takes $O(1)$ time to scan whether an edge exists between any two given vertices, while an adjacency list would take $O(n)$, where n is the number of vertices.

7. Suppose that you want only to find all vertices that are adjacent to some particular vertex. Does an adjacency matrix or an adjacency list provide a more efficient way of doing this?

   An adjacency list is more efficient because finding all the neighbors of a particular vertex only requires traversing the appropriate list, while an adjacency list would need to look through all of the possible edges that could be connected to a particular vertex (traverse an entire row). Both take $O(n)$ time, where n is the number of vertices, but the running time of using an adjacency list may be substantially faster than using an adjacency matrix.

17. A **loop** is an edge that starts and ends at the same vertex. Figure 31-5 shows an example of a loop in a directed, weighted graph.

    (a) Give an example of a problem where allowing loops would be useful.

    Mapping out the love emotion in a group of individuals. The vertices are people. The edges represent love between people. There will be people that love themselves and others, but some that do not. The people that love themselves, will have an edge from and to themselves.

    (b) Can the adjacency matrix and adjacency list representations of a graph support loops?

    Both the adjacency matrix and adjacency list representations of a graph support loops. Adjacency matrices have the entries a_{ij}, where $i = j$ that can be used to represent loops for vertex i. Adjacency lists can just add an edge from one vertex to itself by adding that to the appropriate adjacency list.