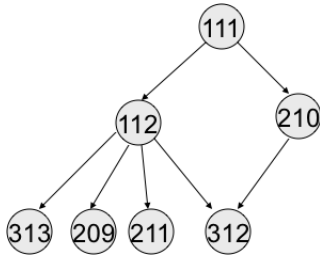


Answers to End of Section and Review Exercises for Chapter 12

Exercises 12.1

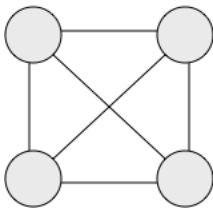
1. Here is the graph.



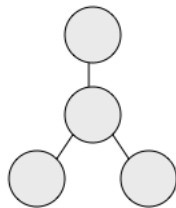
©2014 Cengage Learning®

2. 10.
3. Point-to-point configuration supports linear-time insertions and removals, whereas star supports constant-time insertions and removals. Here are the graphs:

Point-to-point



Star



©2014 Cengage Learning®

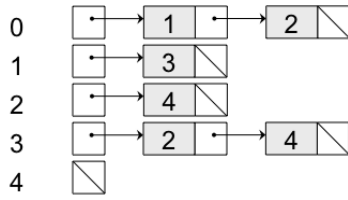
Exercises 12.2

1. Here is the matrix.

	0	1	2	3	4
0	0	1	2	0	0
1	0	0	0	2	0
2	0	0	0	1	3
3	0	0	0	0	2
4	0	0	0	0	0

©2014 Cengage Learning®

2. Here is the picture.



©2014 Cengage Learning®

3. The adjacency matrix supports constant-time searches for an edge between two vertices, whereas the adjacency list supports linear-time (with its edges) searches for an edge. The adjacency list supports linear-time (with its edges) return of neighboring vertices, whereas the adjacency matrix supports linear-time (with all of the vertices in the graph) return of neighboring vertices.

Exercises 12.3

1. A D E B C.
2. A D B E C.
3. A breadth-first traversal of a graph visits all the immediate neighbors of the start vertex. The traversal then visits all the immediate neighbors of these vertices, and so on, until all the vertices reachable from the start vertex have been visited.

Exercises 12.4

1. The algorithm to build and return the shortest path between two vertices first runs the single-source shortest path algorithm with the source vertex. The algorithm then gathers the ancestors of the destination vertex from the grid by adding that vertex's immediate parent to a list, then that parent's parent, and so on, until the source vertex is reached. The contents of the list are then reversed to obtain the desired path.
2. Here are the constant and the functions:

```
INFINITY = "-"
```

```
def isLessWithInfinity(a, b):
    """Returns True if a < b, or False otherwise.
    a and/or b might be INFINITY."""
    if a == INFINITY and b == INFINITY: return False
    elif b == INFINITY: return True
    elif a == INFINITY: return False
    else: return a < b
```

```
def minWithInfinity(a, b):  
    """Returns the smaller of a and b, or either one otherwise."""  
    if isLessWithInfinity(a, b): return a  
    else: return b
```

Answers to Review Questions

1. b
2. b
3. b
4. a
5. b
6. a
7. b
8. c
9. b
10. b