Chapter 22 Graphs and Applications

1. See §22.1.

2. See §22.2.

3. See §22.3.

4.

Edge list:

```
edges = [
  [0, 1], [0, 2], [0, 3], [0, 4], [0, 5],
  [1, 0], [1, 2], [1, 3], [1, 4],
  [2, 0], [2, 1], [2, 3], [2, 4],
  [3, 0], [3, 1], [3, 2], [3, 4], [3, 5],
  [4, 0], [4, 1], [4, 2], [4, 3],
  [5, 0], [5, 3]
]
```

List of edge objects:

```
class Edge:
    def __init__(self, u, v):
        self.u = u
        self.v = v

list = []
list.append(Edge(0, 1))
list.append(Edge(0, 2))
list.append(Edge(0, 3))
list.append(Edge(0, 4))
list.append(Edge(0, 5))
…
…
…
```

Adjacency matrix:

```
adjacencyMatrix = [
  [0, 1, 1, 1, 1, 1], # node 0
  [1, 0, 1, 1, 1, 0], # node 1
  [1, 1, 0, 1, 1, 0], # node 2
  [1, 1, 1, 0, 1, 1], # node 3
  [1, 1, 1, 1, 0, 0], # node 4
  [1, 0, 0, 1, 0, 0]  # node 5
]
```

Adjacency list:

```
list = [[1, 2, 3, 4, 5], [0, 1, 3, 4], [0, 1, 3, 4], [0, 1, 2, 4, 5],
  [0, 1, 2, 3], [0, 3]]
```

5. The Graph class models a graph. The Tree class defines a tree. The Graph contains the method that returns a depth-first tree and a breadth-first tree.

6. dfs(v) returns an object of Tree. bfs(v) returns an object of Tree.

7. A DFS tree for a complete graph is like a straight line connecting all vertices. A BFS tree for a complete graph consists of the root with all other vertices as the children of the root.

8. There are n(n-1)/2 edges in a complete graph. There are n-1 edges in a tree.

9. Omitted
10. There is a possibility that a vertex may be pushed into the stack more than once. Can you give such an example?
11. Omitted
12.

`getIndex("HTHTTTHHH".toCharArray())` returns 184.

`getNode(46)` returns `HHHTHTTTH`