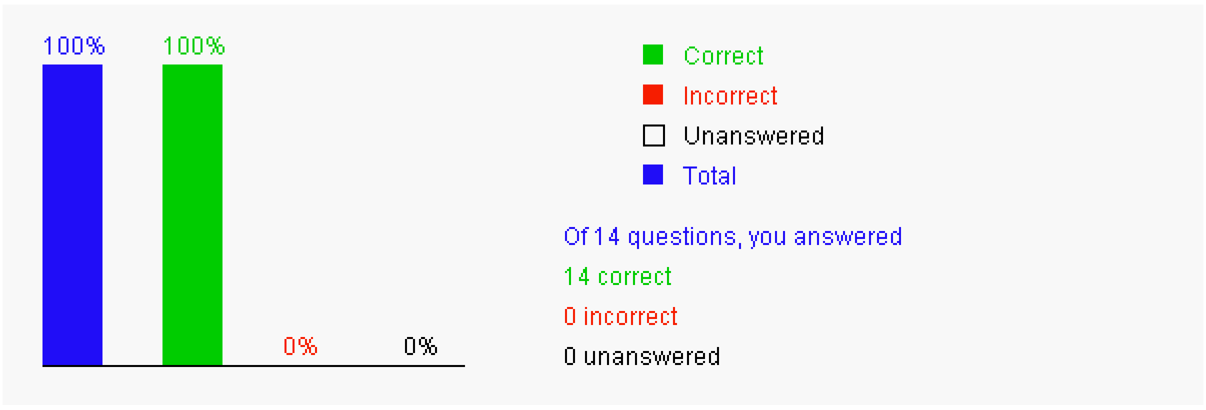
Introduction to Java Programming, Includes Data Structures, Eleventh Edition, Y. Daniel Liang

This quiz is for students to practice. A large number of additional quiz is available for instructors using Quiz Generator from the Instructor's Resource Website. Videos for Java, Python, and C++ can be found at https://yongdanielliang.github.io/revelvideos.html.

Chapter 28 Graphs and Applications



Of 14 questions, you answered
14 correct
0 incorrect
0 unanswered

Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate which book and edition you are using. Thanks!

Section 28.2 Basic Graph Terminologies

28.1 A ____ is an edge that links a vertex to itself.

- [✓ A. loop]
- B. parallel edge
- C. weighted edge
- D. directed edge

Your answer is correct ✓

28.2 If two vertices are connected by two or more edges, these edges are called ______.

- A. loop
- [✓ B. parallel edge]
- C. weighted edge
- D. directed edge

Your answer is correct ✓

28.3 A _________ is the one in which every two pairs of vertices are connected.

- [✓ A. complete graph]
- B. weighted graph
- C. directed graph

Your answer is correct ✓

28.4 What is the number of edges in a complete graph of n vertices?

- A. n
- B. n - 1
- [✓ C. n(n-1)/2]
- D. n*n

Your answer is correct ✓

28.5 What is the number of edges in a tree of n vertices?

- A. n
- [✓ B. n - 1]
- C. n(n-1)/2
- D. n*n

Your answer is correct ✓

Section 28.4 Modeling Graphs

28.6 Suppose a graph is created in the following code. What is the output of the following code?

```
String[] vertices = {"Atlanta", "Dallas", "Chicago", "New York", "Seattle"};
```

```
int[][] edges = {
  {0, 1}, {0, 2},
  {1, 0}, {1, 2}, {1, 3}, {1, 4},
  {2, 0}, {2, 1}, {2, 3},
  {3, 1}, {3, 2}, {3, 4},
  {4, 1}, {4, 3}
};

Graph<String> graph1 = new UnweightedGraph<>(vertices, edges);
System.out.println("The index of vertex Chicago is: "
  + graph1.getIndex("Chicago"));
```

A. 1
[✓ B. 2]
C. 3
D. 4
E. 5

Your answer is correct

28.7 Suppose a graph is created in the following code. What is the number of vertices in the graph?

```
Integer[] vertices = {0, 1, 2, 3, 4};

int[][] edges = {
  {0, 1}, {0, 2},
  {1, 0}, {1, 2}, {1, 3}, {1, 4},
  {2, 0}, {2, 1}, {2, 3},
  {3, 1}, {3, 2}, {3, 4},
  {4, 1}, {4, 3}
};

Graph<Integer> graph1 = new UnweightedGraph<>(vertices, edges);
System.out.println("The number of vertices in graph1: "
  + graph1.getSize());
```

A. 1
B. 2
C. 3
D. 4
[✓ E. 5]

Your answer is correct

28.8 Suppose a graph is created in the following code. What is the degree of vertex 3 in the graph?

```
Integer[] vertices = {0, 1, 2, 3, 4};

int[][] edges = {
  {0, 1}, {0, 2},
  {1, 0}, {1, 2}, {1, 3}, {1, 4},
  {2, 0}, {2, 1}, {2, 3},
  {3, 1}, {3, 2}, {3, 4},
  {4, 1}, {4, 3}
};

Graph<Integer> graph1 = new UnweightedGraph<>(vertices, edges);
System.out.println("The degree of vertex 3: "
  + graph1.getDegree(3));
```

A. 1
B. 2
[✓ C. 3]
D. 4
E. 5

Your answer is correct

Section 28.7 Depth-First Search (DSF)

28.9 Suppose a graph is created in the following code. Using the dfs algorithm in the text, what is the output for the path from 4 to 0?

```
Integer[] vertices = {0, 1, 2, 3, 4};

int[][] edges = {
  {0, 1}, {0, 2},
  {1, 0}, {1, 2}, {1, 3}, {1, 4},
  {2, 0}, {2, 1}, {2, 3},
  {3, 1}, {3, 2}, {3, 4},
  {4, 1}, {4, 3}
};

Graph<Integer> graph1 = new UnweightedGraph<>(vertices, edges);
AbstractGraph<Integer>.Tree dfs = graph1.dfs(0);
System.out.println(dfs.getPath(4));
```

- ○ A. [4, 3, 2, 0]
- ○ B. [4, 3, 1, 0]
- ○ C. [4, 1, 0]
- [✓ D. [4, 3, 2, 1, 0]]
- ○ E. [4, 1, 2, 0]

Your answer is correct ✓

28.10 The _______ search of a graph first visits a vertex, then it recursively visits all the vertices adjacent to that vertex.

- [✓ A. depth-first]
- ○ B. breadth-first

Your answer is correct ✓

28.11 The time complexity of the DFS algorithm is O(|E| + |V|).

- [✓ A. true]
- ○ B. false

Your answer is correct ✓

Section 28.9 Breadth-First Search

28.12 Suppose a graph is created in the following code. Using the bfs algorithm in the text, what is the output for the path from 4 to 0?

```
Integer[] vertices = {0, 1, 2, 3, 4};

int[][] edges = {
  {0, 1}, {0, 2},
  {1, 0}, {1, 2}, {1, 3}, {1, 4},
  {2, 0}, {2, 1}, {2, 3},
  {3, 1}, {3, 2}, {3, 4},
  {4, 1}, {4, 3}
};

Graph<Integer> graph1 = new UnweightedGraph<>(vertices, edges);
AbstractGraph<Integer>.Tree bfs = graph1.bfs(0);
System.out.println(bfs.getPath(4));
```

- ○ A. [4, 3, 2, 0]
- ○ B. [4, 3, 1, 0]
- [✓ C. [4, 1, 0]]
- ○ D. [4, 3, 2, 1, 0]
- ○ E. [4, 1, 2, 0]

Your answer is correct ✓

28.13 The time complexity of the BFS algorithm is O(|E| + |V|).

- [✓ A. true]
- ○ B. false

Your answer is correct ✓

28.14 The _______ search of a graph first visits a vertex, then all its adjacent vertices, then all the vertices adjacent to those vertices, and so on.

- ○ A. depth-first
- [✓ B. breadth-first]

Your answer is correct ✓