True/False

a. The compareTo method compares two strings alphabetically. T
b. Array stores a collection of objects that have identical types. T
c. An ArrayList can shrink automatically. F
d. You can insert an element in the middle of a LinkedList in constant time. F
e. In Java, Queue is an interface. LinkedList implements the Queue. T
f. In quicksort, a pivot divides an array into two sub-arrays. The elements in one of the sub-arrays are smaller than or equal to the pivot while the elements in the other sub-array are larger than or equal to the pivot. T
g. By default, PriorityQueue orders its elements according to their natural ordering. T
h. To find out whether an element exists in a binary search tree, a search method starts the search from a leaf of the tree. F
i. Two objects have the same hash code if they have identical contents. T
j. In max-heap, each node is less than or equal to both of its children. F
k. A graph may have several minimum spanning trees. T

Multiple Choice

1. Suppose current is a reference to a linked list node that has two fields: data and next. Assume that current refers to the last node of the list. Which of the following boolean expressions is true?

   a. current == null
   b. current.next == null
   c. current.data == null

2. The Big-O running time of inserting an element into a balanced binary search tree is O(f). What is f?

   a. The height of the tree.
   b. The number of nodes in the tree.
   c. The number of fields in each node of the tree.

3. Mergesort divides an array into two sub-arrays. Which statement is true before these sub-arrays are merged?

   a. The array elements form a heap.
b. Elements in each sub-array are sorted amongst themselves.
c. Elements in one of the sub-arrays are less than or equal to elements in the other sub-array.

4. What is the compile-time type of an object that is returned from the `iterator` method?

1. Iterator
2. Collection
3. ArrayList

5. Suppose `list1` is an `ArrayList` and `list2` is a `LinkedList`. Assume that `list1` and `list2` contain many values:

```
// Fragment A
for (int i = 0; i < list1.size(); i++) {
    sum = sum + list1.get(i);
}

// Fragment B
for (int i = 0; i < list2.size(); i++) {
    sum = sum + list2.get(i);
}
```

a. Fragment A runs faster than fragment B.
b. Fragment B runs faster than fragment A.
c. Fragment A runs as fast as fragment B.

6. What is the Big-O running time of partitioning an array of N elements in the quicksort algorithm?

a. O(1)
b. O(N)
c. O(log N)

7. Which condition guarantees that a doubly linked list with header and trailer (sentinel) nodes is empty?

a. header == null
b. trailer == null
c. header.next == trailer && trailer.prev == header

8. What is the key "123" mapped to after executing the following program?

```java
public class Test {
    public static void main(String[] args) {
        Map<String, String> map = new HashMap<>();
        map.put("123", "John Smith");
        if (map.get("123") == "John Smith") {
            map.put("123", "George Smith");
        }
    }
}
```
a. "John Smith"
   b. "George Smith"

9. What kind of set is used for storing unique elements in an arbitrary order?
   a. TreeSet
   b. HashSet

10. In which of the following traversals is the node processed after processing its children are complete?
    a. Inorder
    b. Preorder
    c. Postorder

11. Suppose keys 3, 4, 45, 21, 92, 12 are inserted into a binary search tree in this order. What is the inorder traversal of the elements?
    a. 3 4 12 21 45 92
    b. 3 4 45 21 12 92
    c. 12 21 92 45 4 3

12. What is the preorder traversal of the elements in an AVL tree after inserting 3, 4, 45, 21, 92, 12 in this order?
    a. 3 4 12 21 45 92
    b. 3 12 4 92 45 21
    c. 21 4 3 12 45 92

13. The height property of the red-black trees specifies that every path from a node to each null link has the same number of
    a. Black nodes
    b. Red nodes

14. Suppose a max-heap represents the following array: {100, 55, 92, 23, 33, 81}. The parent of 81 is:
    a. 100
    b. 55
    c. 92

15. Which of the following does not use a standard queue?
16. Suppose $a \, R \, b$ indicates that $a$ is related to $b$. Which of the following is an equivalence relation?

   a. $a \, R \, b$ if there is a path from vertex $a$ to vertex $b$ in a directed graph.
   b. $a \, R \, b$ if class $a$ inherits from class $b$.
   c. none of the above

**Coding**

1. Suppose the nodes of a binary tree containing strings are represented by:

   ```java
   class BinaryNode {
       String data;
       BinaryNode left;
       BinaryNode right;
   }
   ```

   An internal node in a binary tree is the one that is not a leaf (it has at least one child). Fill in the following blanks so the `numInternalNodes` method returns the number of internal nodes of a non-empty binary tree. Note that the `numInternalNodes` method is a recursive method.

   ```java
   int numInternalNodes(BinaryNode root) {
       if ( root.left == null && root.right == null ) {
           return 0;
       }
       return 1 + numInternalNodes (root.left) + numInternalNodes (root.right);
   }
   ```

2. The following code uses a `Comparator` to sort an array of Strings in the decreasing order of their lengths. Write the definition of the `Comparator`.

   ```java
   public class SortStringByLength {
       public static void main ( String[ ] args ) {
           String[ ] cities = {"Atlanta", "Savannah", "Dallas", "New York"};
           java.util.Arrays.sort(cities, new LengthComparator( ) );
       }
   }

   class LengthComparator implements Comparator<String> {
       public int compare(String str1, String str2) {
           return str2.length( ) – str1.length ( );
       }
   }
   ```
**ArrayList**

Suppose `list` is an ArrayList of Strings, where each String contains English letters. Write a for-each loop (enhanced for-loop), which iterates through this list and changes the first letter to lowercase for each element in the `list`. Then, it displays the element to the standard output.

```java
for(String element:list) {
    StringBuilder sb = new StringBuilder(element);
    sb.setCharAt(0, Character.toLowerCase(sb.charAt(0)));
    System.out.println(sb.toString());
}
```

**LinkedList**

Suppose a singly circular linked list, which implements a queue. Suppose also that this linked list can have a list iterator that enqueues and dequeues elements at constant time. Does this list iterator need to be initialized to the first element or the last element in the linked list?

**The last element**

**Binary heap**

Draw the binary heap after applying the `remove` operation on the following min-heap. Also, show the array representation of the binary heap using the zero-based array indexing after completing the `remove` operation.

```
7 8 12 9 11 15 14
```

**Hash Map**

Suppose an empty hash map of length 10 and the hash function \( H(k) = k \mod 10 \), where \( k \) is an integer element. The hash map uses the linear probing for resolving collisions. The elements 22, 15, 13, 2, 43, 23, 5 and 18 are inserted into the hash map. Draw the resulting hash map.

```
22 13 2 15 43 23 5 18
```
Trees

1. Consider the following AVL tree:

Assume element 2 is inserted into the above tree.

a. Does it need a single rotation or double rotation to restore the balance condition of the tree?
   Single rotation – 7 is imbalanced. 2 is inserted in left subtree of the left child of 7.

b. Draw the resulting tree.

2. Consider the following red-black tree.

Show the resulting red-black tree after inserting element 4 into the tree. Use the top-down insertion strategy.
Graphs

1. Use the Dijkstra’s algorithm to find the shortest paths from vertex 0 to the other vertices.

What is the last vertex popped from the priority queue? 4
2. Show a topological sorting of vertices in the following directed graph. Note that this graph has more than one topological sorting. *It is enough to show only one of them.*

One of the topological sortings is: V0, V1, V4, V3, V2, V6, V5, V7, V8, V9.

**Disjoint Sets**

Suppose an undirected weighted graph that has the following edges:

<table>
<thead>
<tr>
<th>Source vertex</th>
<th>Destination vertex</th>
<th>Edge weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>F</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>F</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>G</td>
<td>9</td>
</tr>
</tbody>
</table>
a. Draw the graph

b. Draw the minimum spanning tree of the above graph using the Kruskal’s algorithm. The minimum spanning tree is shown by the red edges.