Name:

1. (3 pts) The interface of an ADT indicates _____.
    a. The input and output types of the ADT operations
    b. How to implement the ADT operations
    c. How to store the data in the ADT
    d. All of the above
2. (3 pts) What is a hash code? What did we use them for in this class?

3. (3 pts) A stack is _____ and a queue is _____.
    a. LIFO, FIFO
    b. FIFO, LIFO
    c. FIFO, FIFO
    d. LIFO, LIFO

4. (3 pts) Assuming a linked list is being used, which of the following statements deletes the node that `current` references, assuming `previous` references the node before `current`?

```
a) previous.next = current;
b) current.next = previous;
c) current.next = current.next;
d) previous.next = current.next;
```

5. (3 pts) Assume you have the following two objects that implement the `Comparable` interface: `Number a, b;` Complete the code example that checks if `a` and `b` are equal:

```
if (                                          )
     System.out.println("Equal");
else
     System.out.println("Not Equal");
```

Complete the code example that checks if `a` is less than `b`:

```
if (                                          )
     System.out.println("Less Than");
else
     System.out.println("Not Less Than");
```

6. (3 pts) Assume that the `LinkedQueue` class implements the `QueueInterface` interface, which of the following statements are valid? Highlight **any** of the following that WILL compile and execute successfully:

a) `QueueInterface<Integer> q = new QueueInterface<Integer>();`
b) `QueueInterface<Integer> q = new LinkedQueue<Integer>();`
c) `LinkedQueue<Integer> q = new LinkedQueue<Integer>();`
d) `LinkedQueue<Integer> q = new QueueInterface<Integer>();`

7. (3 pts) _____ resolves collisions by searching for the next available entry in the hash table using second hash function.

a. separate chaining

b. double hashing

c. linear probing

d. no collision resolution is necessary

8. (3 pts) A binary search tree specifies

a. left_child < root < right_child

b. left_child > root > right_child

c. left_child <= root <= right_child

d. left_child < root > right_child

9. (3 pts) What is the worst-case performance of searching for an element in a binary search tree? Hint: How many levels will a tree with n nodes have?

a. O(n)

b. $O(n^2)$

c. O(lg n)

d. O(1)

10. (2 pts) A complete binary tree with 15 nodes has height _____. (Assuming we begin numbering height beginning from 0.)

a) 4

b) 3

c) 15

d) 1

11. (5 points) Order the following growth rates from fastest (most efficient) to slowest (least efficient).)

$O(n \lg n)$ $O(2^n)$ $O(1)$ $O(n)$ $O(n^2)$ $O(\lg n)$

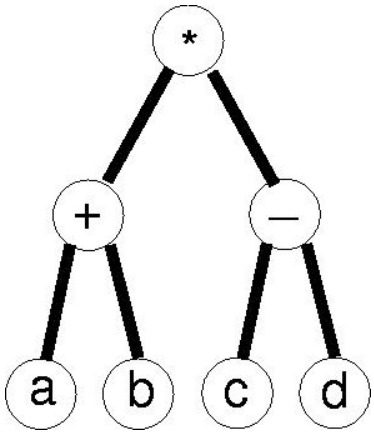
Most efficient Least Efficient

12. (5 pts) Draw the binary search tree that results from inserting the following integers, in the order they appear, into an initially empty binary search tree:

42, 23, 75, 83, 31, 78, 87, 27

13. (3 pts) Using **your** answer to the previous question, delete the root element from the binary search tree, and illustrate what the resulting tree appears as.

14. (6 pts) Given the binary tree



a) What is the pre-order traversal of this tree?

b) What is the in-order traversal of this tree?

c) What is the post-order traversal of this tree?

15. (5 pts) Given the following method, trace the recursive calls to find the result of `mystery(2,25)`. The trace has been started for you.

```
public int mystery(int a, int b) {
     if (b == 0)
          return 0;
     else if (b % 2 == 0)
          return mystery(a + a, b / 2);
     else
          return a + mystery(a + a, b / 2);
}
```

mystery(2,25)
a = 2, b = 25
mystery(4,12)

… Fill this part in…

return 2 + mystery(4,12) = (Fill this in with the correct return value for mystery(2,25)

16. (9 pts) Given the following original array

7	4	3	9	2	6	1	5

What would the array look like after one iteration of the selection sort?

What would the **original** array look like after one iteration of the insertion sort?

Using the right-most element as the pivot, what would the **original** array look like after one iteration of the quick sort? (The answer to this one is flexible. Just think: will be elements to the right of the pivot? What elements are to the left?)

17. (5 pts) Insert the following elements into a dictionary of size 11, using linear probing as a collision resolution mechanism. Assume that the hash code of each number is the number itself. For example, the hash code of 17 is 17.

17, 14, 30, 90, 21, 66, 11, 55

18. (10 pts) For each of the following growth rates, provide an example of an algorithm with that growth rate. The algorithm may be a kind of search (e.g. sequential search, binary search), a kind of sort (e.g. selection sort, merge sort), or a kind of lookup (linked list get, dictionary get).

$O(n)$ $O(n^2)$ $O(\lg n)$ $O(n \lg n)$ $O(1)$

Growth Rate **Algorithm**

a.

b.

c.

d.

e.

19. (10 pts) Which ADT (bag, dictionary, list, priority queue, queue, set, stack, tree) is the most appropriate for each of the following applications? You may use a given ADT more than once. You may state any assumptions you may have made.

a) A list of phone calls to return, ordered by importance.

b) An application that associates a stock symbol (i.e. “GOOG”) to a company name (i.e. “Google”).

c) An algorithm that reverses the letters of a string.

d) A collection of unique names to name your pet.

e) A printer that must print jobs in the order they were received.

20. (5 pts) Assume the class `OrderedList` has a method that returns an Iterator:

```
public Iterator<T> iterator()
```

and you have created the following `OrderedList` object:

```
List<Integer> list = new OrderedList<Integer);
```

Write the code example that iterates through all elements in the `list` object. Use only the methods allowed by the Iterator interface: