CS 300 (Java) Modules 7–9 Practice Test – Answer Key & Explanations

Part 1: Multiple Choice Answers

1. C) O(log n)
Binary search halves the array each step, giving logarithmic complexity.

2. B) Merge Sort
Merge Sort uses recursion and is stable since it preserves order of equal elements.

3. C) Stack overflow error
Without a base case, recursion never stops, overflowing the call stack.

4. C) Time increases roughly by the square of input size
O(n^2) means doubling input quadruples time.

5. B) Binary Search
Binary search is fastest for sorted data (O(log n)).

6. B) Simpler code for divide-and-conquer problems
Recursion simplifies problems like sorting and tree traversal.

Part 2: Trace the Code

7. Output: 3 2 1 1 2 3
Explanation: Each call prints before and after recursion. Sequence prints down from n to 1, then back up.

8. Output: 4 3 2 1 Blastoff!
Explanation: Each call prints n, then calls itself with n-1 until n == 0, then prints 'Blastoff!'.

Part 3: Short Answer

9. Merge Sort vs Quick Sort:

- Merge Sort is **O(n log n)** in all cases and is **stable**.
- Quick Sort averages **O(n log n)** but can degrade to **$O(n^2)$** worst case.
- Quick Sort usually uses less extra memory, while Merge Sort requires temporary arrays.

10. Big O notation measures how an algorithm’s time or space grows with input size.
It’s important because it lets programmers predict performance, compare algorithms, and choose efficient solutions for large data.

Part 4: Free Response Question (FRQ)

11. Recursive method to sum digits:

```
public static int sumDigits(int n) {
if (n < 10) return n; // Base case
return (n % 10) + sumDigits(n / 10); // Recursive case
}
```

Explanation: The base case handles single-digit numbers. The recursive case extracts the last digit (n % 10) and adds it to the sum of the remaining digits.