CS 300 (Java) Practice Test: Modules 7–9

Topics: Algorithm Analysis, Recursion, Sorting & Searching

Part 1: Multiple Choice (1 point each)

1. What is the time complexity of binary search in the worst case?
A) O(n) B) O(n log n) C) O(log n) D) O(1)

2. Which sorting algorithm is both recursive and stable?
A) Quick Sort B) Merge Sort C) Selection Sort D) Heap Sort

3. What will happen if a recursive method has no base case?
A) The compiler throws an error B) The program runs faster C) Stack overflow error D) It converts to a loop

4. Which of the following best describes $O(n^2)$?
A) Time doubles when input doubles B) Time increases exponentially C) Time increases roughly by the square of input size D) Time stays constant

5. Which algorithm is most efficient for a sorted list?
A) Linear Search B) Binary Search C) Selection Sort D) Bubble Sort

6. What is the main advantage of recursion?
A) Always faster than loops B) Simpler code for divide-and-conquer problems C) Uses less memory D) Avoids stack frames

Part 2: Trace the Code (2 points each)

7. What does the following recursive method output when mystery(3) is called?

```
void mystery(int n) {
if (n > 0) {
System.out.print(n + ' ');
mystery(n - 1);
System.out.print(n + ' ');
}
}
```

Output: ___________________________

8. What is printed for countDown(4)?

```
int countDown(int n) {
if (n == 0) {
System.out.print('Blastoff!');
return 0;
}
System.out.print(n + ' ');
return countDown(n - 1);
}
```

Part 3: Short Answer (3 points each)

9. Explain the difference between Merge Sort and Quick Sort in terms of performance and stability.

10. What does Big O notation measure, and why is it important in computer science?

Part 4: Free Response Question (FRQ – 6 points)

11. Recursion FRQ:
Write a recursive Java method called sumDigits that returns the sum of all digits in a positive integer.
Example: sumDigits(532) → 10 (because 5 + 3 + 2 = 10)

```
public static int sumDigits(int n)
```

Include a base case, a recursive call, and a brief explanation of how it works.