### Multiple Choice Questions:

1. Consider the following growth rate function in terms of input size (n): 10, √n, n, log2n, 100n. Which of the following is the correct arrangement of the given growth functions in non-decreasing order of growth rate.

A: log2n, 100n, 10,√n, n

B: 100n, 10, log2n, √n, n

C: 10, 100n, √n, log2n, n

D: 10, log2n, √n, n, 100n

**Answer: D**

2. Consider the following functions:

f(n) = n^2

g(n) = n!

h(n) = nlogn

Which of the following statements about the asymptotic behavior of f(n), g(n), and h(n) is true?

A: f (n) = O(g(n)); g(n) = O(h(n))

B: f (n) = Ω(g(n)); g(n) = O(h(n))

C: g(n) = O(f (n)); h(n) = O(f (n))

D: h(n) = O(f (n)); g(n) = Ω(f (n))

**Answer: D**

3. \_\_\_\_\_\_\_ is the formal way to express the upper bound of an algorithm’s running time.

1. Omega Notation
2. Theta Notation
3. Big Oh Notation
4. All of the above

**Answer: C**

The notation O(n) is the formal way to express the upper bound of an algorithm’s running time. It measures the worst case time complexity or the longest amount of time an algorithm can possibly take to complete.

4. Which of the following is linear asymptotic notations?

1. O(1)
2. O(log n)
3. O(n)
4. O(n log n)

**Answer: C**

Explanation: Linear O(n)

5. Omega notation is the formal way to express the lower bound of an algorithm’s running time:

1. TRUE
2. FALSE
3. Can be true or false
4. Can not say

**Answer: A**

Explanation: True, Omega Notation is the formal way to express the lower bound of an algorithm’s running time.