1. Rank the following functions by order of growth:

    $\lg(\lg n), n^2, e^n, n^3, n, \lg(n), 2^n, n(\lg(n)), 1$

2. Show that for any real constants a and b, where $b > 0$, $(n + a)^b = O(n^b)$

3. What does the following algorithm do (output)? Analyze its worst-case running time, and express it using "Big-Oh" notation.

    Algorithm Foo (a, n):
    Input: two integers a and n
    Output: ?
    k = 0
    b = 1
    while k < n **do**
        k = k + 1
        b = b * a
    return b

4. What does the following algorithm do (What is the output)? Analyze its worst-case running time, and express it using "Big-Oh" notation.

    Algorithm Foo (A, n):
        Input: An array A storing $n \geq 1$ integers.
        Output: ??
    K = A[0]
    for i = 1 **to** n − 1 **do**
        if K > A[i] **then**
            K = A[i]
        return K

5. Determine whether each statement is true or false. Justify your answer
    a) $n^n = O(2^n)$
    b) If $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$
    c) $2^{n+1} = O(2^n)$

6. Show that $\lg n = O(\sqrt{n})$

7. Show that $2n^2 − 3n$ is $\Omega(n^2)$

8. Show that $n^2 + 3n = O(n^2)$

9. Show that $10n^2 − 5n + 51 = o(n^3)$

10. Show that $18n^3 − 4n^2 + 2n − 40 = \omega(n^2)$