/*
* NAME: Megan Chu
* ID: A12814536
* LOGIN: cs12waot
*/

11. For (n)^(1/2) + n, n is of higher order, so it will take precedence as the function increases. Because the function (nlogn) increases at a greater rate than the function (n) as n grows larger, it is impossible to have any C or n0 value that would make n >= C(nlogn).

12. For (100n^3 + n^2), 100n^3 is of higher order, so it will take precedence as the function increases. If we let C = 102, and let n0 be any number >= 1, for example n0 = 1. We can see the following:

100(1)^3 + (1)^2 <= (102)(1)^3
100 + 1 <= 102
101 <= 102, which holds true

At the same time, if we let C = 100, and let n0 be any number >= 1, for example n0 = 1. We can see the following:

100(1)^3 + (1)^2 >= (100)(1)^3
100 + 1 >= 100
101 >= 100, which holds true

Therefore we can say that (100n^3 + n^2) is both big O and big omega of (n^3). The functions are of the same order, therefore it is true that (100n^3 + n^2) is big theta of (n^3) (there is both an upper and lower bound).

13. If we simplify the left function we get:

(1/n) + (n^2/n)
n^(-1) + n, because n is of higher order than n^(-1), n takes precedence as the function increases

If we let C = 3, and let n0 be some number >= 1, for example n0 = 1, we see the following:

(1/(1)) + ((1)^2/(1)) <= 3((1))
1 + 1 <= 3
2 <= 3, which holds true

At the same time, if we let C = 1, and let n0 be some number >= 1, for example n0 = 1, we see that:

(1/(1)) + ((1)^2/(1))) >= 1((1))
1 + 1 >= 1
2 >= 1, which holds true

Therefore we can say that (1/n + n^2/n) is both big O and big omega of (n). The functions are of the same order, so it holds true that (1/n + n^2/n) is big theta of (n) (there is both an upper and lower bound).

14. For (1/n^100) + log16, we can rewrite this as n^(-100) + (n^0)log16. Because (n^0)log16 is of higher order than n^(-100), it takes precedence as the function increases. If we let C = (log16 + 2), and let n0 be some number >= 1, for example n0 = 1, we can see that:

1/(1)^100 + log16 <= (log16 + 2)(1)
1 + log 16 <= log16 + 2, which holds true

At the same time, if we let C = log 16, and let n0 be some number >=1, for example n0 = 1, we can see that:

1/(1)^100 + log16 >= (log16)(1)
1 + log16 >= log16, which holds true

Therefore, we can say that (1/n^100 + log16) is both big O and big omega of (1). Because both functions are of the same order, it is true that (1/n^100 + log16) is big theta of (1) (there is both an upper and lower bound).

15. For the left function (n + log(n^2)) n grows at a faster rate and takes precedence as the function increases. If we let C = 1, and let n0 equal some number >= 1, for example n0 = 1, we find the following:

(1) + log(1)^2 <= (1)(5(1))
1 + 0 <= 5
1 <= 5, which holds true

At the same time, if we let C = 1/10, and let n0 equal some number >= 1, for example n0 = 1, we find the following:

(1) + log(1)^2 >= (1/10)(5(1))
1 + 0 >= 1/2
1 >= 1/2, which holds true

Therefore, we can say that it is true that (n + logn^2) is both big O and big omega of (5n). The functions are of the same order, so therefore (n + logn^2) is big theta of (5n) (there is both an upper and lower bound).