1. For each of the below GIVEN statements, state whether the following OTHER statement is definitely true, definitely false, or possible (i.e., it could be true or false).
2. GIVEN: T1(N) = O(N2)

OTHER: T1(N) = O(N5)

**TRUE**

1. GIVEN: T1(N) = O(N5)

OTHER: T1(N) = O(N2)

**POSSIBLE**

1. GIVEN: T1(N) = Ω(N2)

OTHER: T1(N) = O(N5)

**POSSIBLE**

1. GIVEN: T1(N) = Θ(N5)

OTHER: T1(N) = O(N2)

**FALSE**

1. GIVEN: T1(N) = O(N5) and T2(N) = O(N3)

OTHER: T1(N) + T2(N) = O(N7)

**TRUE**

1. GIVEN: T1(N) = O(N5) and T2(N) = O(N3)

OTHER: T1(N) + T2(N) = Θ(N4)

**FALSE**

1. GIVEN: T1(N) = Θ(N5) and T2(N) = Θ(N3)

OTHER: T1(N) + T2(N) = Ω(N4)

**FALSE**

1. GIVEN: T1(N) = o(N2) and T2(N) = Ω(N)

OTHER: T1(N) + T2(N) = Θ(N2)

**FALSE**

1. GIVEN: T1(N) = o(N2) and T2(N) = O(N)

OTHER: T1(N) + T2(N) = Θ(N2)

**FALSE**

1. GIVEN: T1(N) = Ω(N3 \* logN) and T2(N) = Θ(N3.5)

OTHER: T1(N) + T2(N) = Ω(N5)

**FALSE**

1. Consider the following pseudo-code for a function that takes, as input, a parameter N:

Function F(N)

Loop i from 1 to N3

if F1(N)

Loop j from 1 to N2

F2(N)

else

Loop j from 1 to N

F3(N)

Loop j from 1 to N

F4(N)

F5(N)

For each of the following assumptions, analyze the worst-case running time of Function F using Big-Oh notation, Big-Omega notation, and Big-Theta notation, *if possible*. Express all answers using the tightest possible bounds. Explain your answers!

Note: In class, we went over rules related to Big-Oh notation. You will have to infer similar rules for the other notations. Assume that function F1 returns a Boolean value.

1. Assume that the worst-case running time of F1(N) is O(N), the worst-case running time of F2(N) is O(N), the worst-case running time F3(N) is O(N), the worst-case running time of F4(N) is O(N), and the worst-case running time of F5(N) is O(N).

One can extrapolate this problem into an equation of the form N3(F1 +F2||F3)+N(F4+F5).

Substituting N for each function, we can simplify the equation to N4+N6+N2+N2. Choosing the worst case in the if statement will yield the highest exponent being N6. In the best case scenario, each linear function could take only a single operation but we know that it will have to loop at least N5 times because of the out N3 loop and inner N2  loop. Therefore, the best running time is N5. Because the best and worst running times are not the same, there is no Big-Theta.

Answer: O(F(N)) = N6 Ω(F(N)) = N5

1. Assume that the worst-case running time of F1(N) is O(N), the worst-case running time of F2(N) is O(N3), the worst-case running time F3(N) is O(N5), the worst-case running time of F4(N) is O(N7), and the worst-case running time of F5(N) is O(N9).

One can use the same equation from above which simplifies this problem to the equation N4+N8||N9+N8+N10. The highest exponent is 10, so the worst case running time is N10. Similar to last time, the best case of each equation is not known, but we know that it must loop at least N5 times for the same reason (the N3 and N2 loops). Therefore the best case running time must be N5.

Answer: O(F(N)) = N10 Ω(F(N)) = N5

1. Assume that the worst-case running time of F1(N) is Ω(N), the worst-case running time of F2(N) is Ω(N), the worst-case running time F3(N) is Ω(N), the worst-case running time of F4(N) is Ω(N), and the worst-case running time of F5(N) is Ω(N).

In this problem, the equation will simplify to the same thing as in (a): N4+N6||N5+N2+N2. The best case of this equation would be N2, but the previous section of the code must run, so N5 is the lowest value that could happen (N4 is only the test value, which causes the N6||N5 section to run). However, we don’t now anything about the worst possible running time. Each function may take an infinite number of operations.

Answer: Ω(F(N)) = N5 O(F(N)) = ∞

1. Assume that the worst-case running time of F1(N) is Θ(N), the worst-case running time of F2(N) is Θ(N), the worst-case running time F3(N) is Θ(N), the worst-case running time of F4(N) is Θ(N), and the worst-case running time of F5(N) is Θ(N).

Simplifying the above equation yields: N4+N6||N5+N2+N2. The worst case of this is N6. The best case that could actually execute given the code snippet is N5. Both N2 snippets can only happen after the N5 code has run so it must be at least N5.

Answer: O(F(N)) = N6  Ω(F(N)) = N5

1. Assume that the worst-case running time of F1(N) is Θ(N8), the worst-case running time of F2(N) is Θ(N3), the worst-case running time F3(N) is Θ(N5), the worst-case running time of F4(N) is Θ(N7), and the worst-case running time of F5(N) is Θ(N9).

Simplifying the above equation yields N11+N8||N9+N8+N10. The worst case of this code is N11. Even though this is in the if statement, it still must run so it is the longest possible running time. The best case is also N11 because the if statement always executes. Because both the worst and best running times are the same, there exists a Big-Theta.

Answer: O(F(N)) = N11 Ω(F(N)) = N11 Θ (F(N)) = N11