Instructions: This homework assignment focuses on algorithm analysis. Submit your answers via the Curator as Quiz: Analysis.

For questions 1 through 5, apply the rules given in the course notes and in class to determine the best big-O classification of the given function. All logarithms are base 2.

1. $f(n) = 4n^2 + 97n*\log n + 1000$ is $O(\_\_\_\_\_)$

1) $O(1)$	4) $O(n \log n)$	7) $O(2^n)$
2) $O(\log n)$	5) $O(n^2)$	8) None of these
3) $O(n)$	6) $O(n^2 \log n)$	

2. $f(n) = 17n + 43n*\log n + \log n$ is $O(\_\_\_\_\_)$

1) $O(1)$	4) $O(n \log n)$	7) $O(2^n)$
2) $O(\log n)$	5) $O(n^2)$	8) None of these
3) $O(n)$	6) $O(n^2 \log n)$	

3. $f(n) = 3n + 100{,}000$ is $O(\_\_\_\_\_)$

1) $O(1)$	4) $O(n \log n)$	7) $O(2^n)$
2) $O(\log n)$	5) $O(n^2)$	8) None of these
3) $O(n)$	6) $O(n^2 \log n)$	

4. $f(n) = 3n(\log n + 7n)$ is $O(\_\_\_\_\_)$

1) $O(1)$	4) $O(n \log n)$	7) $O(2^n)$
2) $O(\log n)$	5) $O(n^2)$	8) None of these
3) $O(n)$	6) $O(n^2 \log n)$	

5. $f(n) = 1000n^2 + 0.001*2^n$ is $O(\_\_\_\_\_)$

1) $O(1)$	4) $O(n \log n)$	7) $O(2^n)$
2) $O(\log n)$	5) $O(n^2)$	8) None of these
3) $O(n)$	6) $O(n^2 \log n)$	

---

6. Using the rules given in the notes and class, Analyze the following code fragment to estimate T(N).

```
int i, Sum;
i = N;
Sum = 0;
while (i > 0) {
   for (int j = 0; j < i; j++)
      Sum++;
   cout << Sum << end;
   i--;
}
```

Based on your analysis, what is the best big-O classification for T(N)?

1) $O(1)$	3) $O(N)$	5) $O(N^2)$
2) $O(\log N)$	4) $O(N \log N)$	6) None of these

7. Using the rules given in the notes and class, Analyze the following code fragment to estimate T(N).

```
int Sum;
Sum = 0;
for (int i = 1; i <= N; i++) {      //Hint: you may assume N is a power of 2
   for (int j = 1; j <= N; j = 2*j)
      Sum++;
}
```

Based on your analysis, what is the best big-O classification for T(N)?

1) O(1)
2) O(log N)
3) O(N)
4) O(N log N)
5) $O(N^2)$
6) None of these

For questions 8 through 10, consider the problem of performing a search on a sorted linked list containing N nodes.

8. In the best case, searching a sorted linked list containing N nodes would be O(____) for comparisons.

1) O(1)
2) O(log N)
3) O(N)
4) O(N log N)
5) $O(N^2)$
6) None of these

9. In the worst case, searching a sorted linked list containing N nodes would be O(___) for comparisons.

1) O(1)
2) O(log N)
3) O(N)
4) O(N log N)
5) $O(N^2)$
6) None of these

10. In the average case, assuming that each element of the list is equally likely to be targeted, searching a sorted linked list containing N nodes would be O(__) for comparisons.

1) O(1)
2) O(log N)
3) O(N)
4) O(N log N)
5) $O(N^2)$
6) None of these

---

11. Suppose that the execution of a particular algorithm requires carrying out T(N) operations, where N is the number of inputs that must be processed and

$$T(N) = 2N^2 + 7N/2 + 42$$

Assume the algorithm will be executed on hardware capable of performing 10^6 operations per second. What is the largest size problem (value of N) for which the algorithm can be carried out in 60 seconds? (Pick the closest answer.)

1) 1
2) 5000
3) 10,000
4) 15,000
5) 20,000
6) 25,000
7) 30,000
8) 40,000
9) 50,000
10) 100,000

12. Suppose that the execution of a particular algorithm requires carrying out T(N) operations, where N is the number of inputs that must be processed and

$$T(N) = N \log N + 7N + 42$$

Assume the algorithm will be executed on an input of size 2^{20}, using the same hardware described in question 11. How long will it take (in seconds)? (Pick the closest answer.)

1) 1
2) 10
3) 20
4) 30
5) 40
6) 50
7) 60
8) 120
9) 1000
10) 10000

13. Repeat question 12, assuming the input is twice as large.

1) 1
2) 10
3) 20
4) 30
5) 40
6) 50
7) 60
8) 120
9) 1000
10) 10000

14. Suppose that the execution of a particular algorithm requires carrying out T(N) operations, where N is the number of inputs that must be processed and

$$T(N) = N^2 + 2N + 1$$

Assume the algorithm will be executed on an input of size 2^{20}, using the same hardware described in question 11. How long will it take? (Pick the closest answer.)

1) 10 minutes
2) 20 minutes
3) 60 minutes
4) 6 hours
5) 12 hours
6) 1 day
7) 10 days
8) 20 days
9) 50 days
10) 100 days

15. Repeat question 14, assuming the input is twice as large.

1) 10 minutes
2) 20 minutes
3) 60 minutes
4) 6 hours
5) 12 hours
6) 1 day
7) 10 days
8) 20 days
9) 50 days
10) 100 days