Name: Liam Brew Date: 02.07.2020

Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Liam Brew

Give the complexity of the following functions. Choose the most appropriate notation from among O, θ, and Ω.

1.
```
void function1(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = i; j <= n; j += 2) {
            cout << "*";
        }
    }
}
```
Answer: $\theta(n*\lg(n))$

2.
```
void function2(int n) {
    int count = 0;
    for (int i = 1; i * i <= n; i++) {
        count++;
    }
    cout << count;
}
```
Answer: $\theta(\sqrt{n})$

3.
```
void function3(int n) {
    int count = 0;
    for (int i = n/2; i <= n; i++) {
        for (int j = 1; j + n/2 <= n; j++) {
            for (int k = 1; k <= n; k *= 2) {
                count++;
            }
        }
    }
    cout << count;
}
```
Answer: $\theta(n^2 * \lg(n))$

4.
```
void function4(int n) {
    int count = 0;
    for (int i = n/2; i <= n; i++) {
        for (int j = 1; j <= n; j *= 2) {
            for (int k = 1; k <= n; k *= 2) {
                count++;
            }
        }
    }
    cout << count;
}
```
Answer: $\theta(n * \lg(n) * \lg(n))$

5.
```
void function5(int n) {
    if (n % 2 == 0) {
        return;
    }
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            cout << "*";
            break;
        }
    }
}
```
Answer: O(n)

6.
```
void function6(int n) {
    int count = 0;
    for (int i = 1; i <= n/2; i++) {
        for (int j = 1; j <= n/3; j++) {
            for (int k = 1; k <= n/4; k++) {
                count++;
            }
        }
    }
    cout << count;
}
```
Answer: $\theta(n^3)$

7.
```
void function7(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j += i) {
            cout << "*";
        }
    }
}
```
Answer: $\theta(n * \log(n))$

8.
```
void function8(int n) {
    int i = 1, s = 1;
    while (s <= n) {
        i++;
        s += i;
        cout << "*";
    }
}
```
Answer: $\theta(\sqrt{n})$

9. Processing Arrays
    a. Suppose you have an unsorted array of integers of length n and want to sum all the elements inside it. What is the running time of your algorithm? $\theta(n)$
    b. Suppose you have an unsorted array of integers of length n and want to determine if all the values inside are positive. What is the running time of your algorithm? $\theta(n)$

c. Suppose you have a sorted array of integers of length n and want to determine the median value. What is the running time of your algorithm? $\theta(c)$

10. T T / F $f(n) = 3n^2 + 4n + 2 \in \theta(n^2)$
If true, prove it by giving *integral* values for the required constants c_1, c_2, and n_0. Choose the tightest values possible for the c_1 and c_2 constants. If false, show the contradiction.

Upper Bound: $3n^2 + 4n + 2 <= 3n^2 + 5n$ for all $n >= 1$ → $3n^2 + 5n <= 4n^2$ for all $n >= 5$ → $c_1 = 4$
Lower Bound: $3n^2 + 4n + 2 >= 3n^2$ for all $n >= 1$ → $c_2 = 3$
Therefore $n_o = \max(n_1, n_2) = \max(1, 5) = n_0 = 5$