

Big O / Worst Case Equation

Checking Your Understanding

Objective: To understand runtime complexity (algorithm efficiency).

Topics	Exercises
Complexity Classes	Order of N, O(?)
Worst Case Function	f(n)

Useful mathematical summations:

$$1 + 2 + 3 + \dots + (n - 1) + n = \frac{n(n + 1)}{2}$$

$$a^0 + a^1 + a^2 + a^3 + \dots + a^{(n-1)} + a^n = \frac{a^{n+1} - 1}{a - 1}$$

$$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{(n-1)} + 2^n = \frac{2^{n+1} - 1}{2 - 1}$$

NB: DO NOT JUST WRITE THE ANSWER. You must explain in some way by (a) a diagram, (b) showing the order of each line of code in braces and finding the dominant term, (c) a graphical representation or (d) a few sentences. **Failure to do so, will result in a reduced score.**

Algorithm Efficiency

1. Consider the algorithm for finding the maximum number in an array of n elements. What is the order of n , $O(?)$ for this algorithm? Explain.

2. What is the order of n , $O(?)$ for the following nested loop?

```
for (i = 1; i <= n; i++) {  
    for (j = 1; j <= i; j++) {  
        k = k + i + j;  
    }  
}
```

3. What is the order of n , $O(?)$ for the following nested loop?

```
for (i = 1; i <= n; i++) {  
    for (j = 1; j <= 20; j++) {  
        k = k + i + j;  
    }  
}
```

4. What is the order of n , $O(?)$ for the following sequence

```
for (j = 1; j <= 10; j++) {  
    k = k + 4;  
}  
  
for (i = 1; i <= n; i++) {  
    for (j = 1; j <= 20; j++) {  
        k = k + i + j;  
    }  
}
```

5. What is the order, $O(?)$ of the following function?

$$n^3 + 100n^2 + n$$

6. What is the order, $O(?)$ of the following function?

$$2^n + 100n^2 + 45n$$

7. Suppose an algorithm takes exactly the given number of statements for each value below, in terms of the size of n , i.e. the order of n , $O(?)$. Explain.

$$n \log n + \log n + n$$

8. Suppose an algorithm takes exactly the given number of statements for each value below, in terms of the size of n , i.e. the order of n , $O(?)$. Explain.

$$n^2 \log n + 2n$$

Find the worst-case runtime $f(n)$ for the following algorithms.

- Specify the number of operations executed for an input size n , for the worst case run time as a function of n .
- Circle statement(s) and draw a line to the right side specifying the number of operations.
- If statement(s) are a part of an iteration of n , specify the total number of iterations as a function of n .

9. Algorithm-01

```
int sum = 0;

int j = 1;

while (j <= n) {

    sum++;

    j = j * 2;

}
```

10. Algorithm-02

```
int sum = 0;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        sum++;
    }
}

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= 20; j++) {
        sum--;
    }
}
```