

Big O Notation

Checking Your Understanding

Objective: To understand runtime complexity (algorithm efficiency).

Topics	Exercises
Complexity Classes	Order of N, O(?)

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.**

Run Time Complexity/Algorithm Efficiency

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

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

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

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

3. What is the order of n , $O(f(n))$ of the following function?

$$n^3 + 20n^2 + 50n$$

4. What is the order, $O(f(n))$ of the following function?

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

5. 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 , $O(f(n))$. Explain.

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

6. 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 , $O(f(n))$. Explain.

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