Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. Which has the largest time to compute? 1 / 1 point

- `O(1)`
- `O(log n)`
- (selected) `O(N)`

Correct
That's correct! This is known as linear time. As the input increases so does the time to compute an output.

2. Given the following lines of code pseudocode; 1 / 1 point

```
N = 7

FOR i = 1 TO N:

    output(i)
```

- `O(n^2)`
- `O(1)`
- (selected) `O(N)`

Correct
That's correct! As the loop is set to the size of N, when N increases so does the time complexity.

3. Given the following lines of code pseudocode; 1 / 1 point

```
N = 7

FOR i = 1 TO N:

    FOR j = 1 TO N:

        output(N)
```

- `O(N)`
- (selected) `O(n^2)`
- `O(1)`

Correct
That's correct. There are 2 loops so every time the application runs, it must do N*N executions.

4. Given the following lines of code pseudocode: 1 / 1 point

```
N = 37

FOR i = 1 TO N:

    j = 1

    WHILE j < 10:

        output(j*N)

        j = j + 1
```

- `O(n^2)`
- `O(1)`

- (●) `O(N)`

Correct
That's correct. The inner loop is only run a finite number of times that does not increase with N.

5. Given the following lines of code pseudocode: **1 / 1 point**

```
N = 10

FOR i = 1 TO 5:

        FOR j = 1 TO i:

                output(i*j)
```

- ( ) `O(n^2)`
- (●) `O(1)`
- ( ) `O(Log N)`

Correct
That's correct. As I is limited to 5. Regardless of how large the input becomes it will always be limited to the number of executions.

6. Given the following lines of code pseudocode: output(N) **1 / 1 point**

```
N = 7

FOR i = 1 TO N:

           FOR j = 1 TO N:

                output(N)
```

- (●) `O(n^2)`
- ( ) `O(N)`
- ( ) `O(1)`

Correct
That's correct. There are 2 loops so every time the application runs, it must do N*N executions.