How to measure the codes using Big O

5 Rules

No	Description	Complexity
Rule 1	Any assignment statements and if statements that are executed once regardless of the size of the problem	O(1)
Rule 2	A simple “for” loop from 0 to n (with no internal loops)	O(n)
Rule 3	A nested loop of the same type takes quadratic time complexity	$O(n^2)$
Rule 4	A loop, in which the controlling parameter is divided by two at each step	O(log n)
Rule 5	When dealing with multiple statements, just add them up	

You should be warned that some declarations may include initializations and some of these may be complex enough to factor into the efficiency of an algorithm.

Python range() Function

Definition and Usage

The **`range()`** function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

Syntax

```
range(start, stop, step)
```

Parameter Values

Parameter	Description
start	Optional. An integer number specifying at which position to start. Default is 0
stop	Required. An integer number specifying at which position to end.
step	Optional. An integer number specifying the incrementation. Default is 1

Examples

Example 1

Create a sequence of numbers from 0 to 6, and print each item in the sequence:

```
for n in range(7):
    print(n)
```

The output is:

```
0
1
2
3
4
5
6
```

Example 2

Create a sequence of numbers from 3 to 6, and print each item in the sequence:

```
for n in range(3,7):
    print(n)
```

The output is:

```
3
4
5
6
```

Example 3

Create a sequence of numbers from 1 to 6, but increment by 2 and print each item in the sequence:

```
for n in range(1,7,2):
    print(n)
```

The output is:

```
1
3
5
```