O(N)	O(∞)	O(NM)	$O(\sqrt{N})$	O(5)	$O(N^2)$	$O(N^{1.5})$
O(log N)	O(N log N)		O(0)	$O(N^4)$	O(2/N)	$O(2^N)$

In terms of Run Time

O(0) < O(2/N) < O(log N) < O(N log N) < O(N^4) < O(2^N)

- What is the complexity of each of the following pieces code? Show work.

i.
```
sum = 0;
for (i = 0; i < n; i++) {
    sum++;
}
```
Loop runs N+1 times so complexity O(N)

ii.
```
sum = 0;
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        sum++;
    }
}
```
An N+1 Loop inside an N+1 loop is effectively an N^2 + 2n + 1 run time, with complexity O(N^2)

iii.
```
sum = 0;
for (i = 0; i < n; i++) {
    for (j = 0; j < i; j++) {
        sum++;
    }
}
```
Outter Loop runs N+1 times, inner loop runs up to the value of N, creating a total factorial behavior with a complexity of O(N!)

iv.
```
sum = 0;
for (i = 0; i < n * n; i++) {
    for (j = 0; j < n * n; j++) {
        sum++;
    }
}
```
N^2 Loop inside an N^2 Loop produces an N^4 loop which is complexity O(N^4)