3. Big-O notation

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

- i.

```
sum=0;          // 1 time
for (i=0; i<n; i++)   // n+1 times; n times
(n times)       // 1 time
    sum++;
```

$2n+2$

$+\ n$

$T(n) = 3n+2$

★ Complexity is O(n)

Not Necessary

- $c \cdot f(n) > T(n)$ for all $n > n_0$
- $c \cdot n > 3n+2$
- $c > 3 + \frac{2}{n}$; set $n_0 = 1$
- $c = 3 + \frac{2}{1} = 5$
- $5 \cdot f(n) > T(n)$ for all $n > 1$

ii.

```
for(i=0; i<n; i++)      // 1 time; n+1 times; n times
(n times)
    for(j=0; j<n; j++)  // 1 time; n+1 times; n times
    (n^2 times)
        sum++;          // 1 time
```

$2n+2$

$2n^2+2n$

$+\ n^2$

$3n^2+4n+2$

★ Complexity is $O(n^2)$