- start

```
i=1;
k=1;
while (k<n)
{
    stmt;
    k=k+i;
    i++;
}
```

end $O(\sqrt{n})$

- start

```
while (m!=n)
    if (m>n)
        m=m-n;
    else
        n=n-m;
```

end

min O(1)
Max O(n)

- Algorithm Test (n)
start

```
if (n<5)
    printf("%d", n);      - (1) Best O(1)
else
    for(i=0; i<n; i++)
        printf("%d", i);  - n worst O(n)
```

end

- start

```
if(n>5)
{
  for(i=0; i<n; i++)
  { printf("%d", i); - n
  }
}
```

end

→ Types of Functions

$O(1)$	→	Constant
$O(\log n)$	→	Logarithemic
$O(n)$	→	Linear
$O(n^2)$	→	Quadratic
$O(n^3)$	→	Cubic
$O(2^n)$	→	Exponential

→ Comparison of types of Function

← lower Bound →

$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 \cdots 2^n < 3^n < n^n$

average Bound ← upper Bound →

Prof:

$\log n$	n	n^2	2^n
0	1	1	2
1	2	4	4
2	4	16	16
3	8	64	256
3.1	9	81	512

→ Asymptotic Notations

O	big-oh	Upper Bound
Ω	bigomega	Lower Bound
θ	theta	Average Bound

- Big-Oh Notation

The function $f(n) = O(g(n))$ iff $\exists$ +ve constant C and N_0 such that $f(n) \leq C * g(n)$ $\forall$ $n \geq n_0$.

eg: $f(n) = 2n+3$ $\therefore f(n) = O(n)$

$2n+3 \leq 10n$ $n \geq 1$

$f(n)$ $C\ g(n)$

- Omega Notation

The function $f(n) = \Omega(g(n))$ iff $\exists$ +ve constant c and n_0 such that $f(n) \geq c * g(n)$ $\forall$ $n \geq n_0$

eg. $f(n) = 2n+3$

$2n+3 \geq 1 \times n$ $\forall$ $n \geq 1$ $\therefore f(n) = \Omega(n)$

$f(n) = \Omega(\log n)$

↓ ↓ ↓

$f(n)$ $c\ g(n)$

- Theta Notation

The function $f(n) = \Theta(g(n))$ iff $\exists$ +ve constant c_1, c_2 and n_0 such that $c_1 * g(n) \leq f(n) \leq c_2 * g(n)$

eg. $f(n) = 2n+3$ $\therefore f(n) = \Theta(n)$

$1 \times n \leq 2n+3 \leq 5 \times n$

$c_1 g(n)$ ↓ $c_2 g(n)$

$f(n)$

- Examples of Asymptotic Notations

1. $f(n) = 2n^2 + 3n + 4$

$2n^2+3n+4 \leq 2n^2 + 3n^2 + 4n^2$

$2n^2 3n+4 \leq 9n^2$ $n \geq 1$

$f(n) = O(n^2)$

↑ ↓

$c\ g(n)$

$2n^2+3n+4 \geq 1 \times n^2 \quad \Omega(n^2)$

$1 \times n^2 \leq 2n^2+3n+4 \leq 9n^2 \quad \Theta(n^2)$

2. $f(n) = n^2 \log n + n$

$1n^2 \log n \leq n^2 \log n + n \leq 10n^2 \log n$

$O(n^2 \log n) \quad \Theta(n^2 \log n) \quad \Omega(n^2 \log n)$

* $n^2 \log n$ lies b/w n^2 and n^3 in function types table

3. $f(n) = n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$

$1 \times 1 \times 1 \times \cdots \times 1 \leq 1 \times 2 \times 3 \times \cdots \times n \leq n \times n \times n \times \cdots \times n$

$1 \leq n! \leq n^n$

$\Omega(1) \quad O(n^n)$

↳ class is not there for $n!$

4. $f(n) = \log n!$

$\log(1 \times 1 \times \cdots \times 1) \leq \log(1 \times 2 \times 3 \times \cdots \times n) \leq \log(n \times n \times \cdots \times n)$

$1 \leq \log n! \leq \log n^n = n \log n$

$\Omega(1) \quad O(n \log n)$

→ Properties of Asymptotic Notations

- General Property
  - if $f(n)$ is $O(g(n))$ then $a * f(n)$ is $O(g(n))$
    eg. $f(n) = 2n^2 + 5$ is $O(n^2)$
    then $7 * f(n) = 7(2n^2 + 5)$
    $= 14n^2 + 35$

- Reflexive property
  - if $f(n)$ is given then $f(n)$ is $O(f(n))$
    eg. $f(n) = n^2$ $O(n^2)$

- Transitive property
  - if $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$ then $f(n) = O(h(n))$
    eg. $f(n) = n$ $g(n) = n^2$ $h(n) = n^3$

    n is $O(n^2)$ and n^2 is $O(n^3)$
    then n is $O(n^3)$

- Symmetric Property
  - if $f(n)$ is $O(g(n))$ then $g(n)$ is $O(f(n))$
    eg. $f(n) = n^2$ $g(n) = n^2$
    $f(n) = O(n^2)$
    $g(n) = O(n^2)$

- Transpose symmetric
  - if $f(n) = O(g(n))$ then $g(n)$ is $\Omega(f(n))$
    eg. $f(n) = n$ $g(n) = n^2$
    then n is $O(n^2)$ and
    n^2 is $\Omega(n)$

when a function is both upper Bound and lower Bound

- if $f(n) = O(g(n))$
  and $f(n) = \Omega(g(n))$

$$g(n) \leq f(n) \leq g(n)$$

$$\therefore f(n) = \Theta(g(n))$$

- if $f(n) = O(g(n))$
  and $d(n) = O(e(n))$
  then $f(n) + d(n) = O(\max(g(n), e(n)))$
  eg: $f(n) = n = O(n)$
  $d(n) = n^2 = O(n^2)$
  $f(n) + d(n) = n + n^2 = O(n^2)$

- if $f(n) = O(g(n))$
  and $d(n) = O(e(n))$
  then $f(n) * d(n) = O(g(n) * e(n))$
