DSA

→ Infinite algorithm's time complexity can't be measured

Efficiency => time
              ∪
              memory occupation

Methods for time complexity

→ Any method which relys on past the machine is invalid (machines could have varying speeds)

prior method (approximation) =>

→ Input size matters e.g

```
for(int i=0; i<4; i++)
{ cout << "Wassap"; }
```

If ~~one~~ single ~~cycle~~ execution of command takes 1 sec than $f(n) = 4$

```
int a=0;
cin >> a;
                    n
for (int i=0; i<9; i++)
{  a= a+1;}
```

$f(n) = n+2$

~~for~~ $f(n) = 11$

```
for (int i = 0; i<4; i++)
{   for(int j=0; j<4; j++)
    {  cout<< "hi";}
```

$f(n) = n^2$

in this case $f(n) = 16$

Growth rate of n : (Assymtotic complexity)

$f(n) = 3n^2 + 8n$

If $n = 1$

% due to $3n^2 = 25\%$ app
% due to $8n = 75\%$ app

Let $n = 10$

% $3n^2 = 79\%$
% $8n = 21\%$

Let $n = 100$

% $3n^2 \doteq 97\%$ app
% $8n = 3\%$ app

Big O	Big omega	Big theta
$O()$	$\Omega()$	$\Theta()$

→ upper bound (least)

Big O (worst case scenario)

$f(n) = O(g(n))$

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

$c > 0$

$n \geq k$

$k \geq 1$

e.g

$f(n) = 3n^2 + n$

$3n^2 + n \leq c \cdot g(n)$

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

$n \geq 4/3$