Ex1

Time | Space

```
swap(a,b)
{
    temp = a;  → 1
    a = b;     → 1
    b = temp;  → 1
}
```

$\boxed{f(n) = 3}$ ↑ Constant

O(1)

a → 1

b → 1

temp → 1

$\boxed{f(n) = 3}$ ↑ words

O(1) Constant

$x = 5*a + 6*b$ → 1 ms

$x = 5*a + 6*b$ → 1 ms

$x = 5*a + 6*b$ → 1 ms

$x = 5*a + 6*b$ → 1 ms

$\boxed{f(n) = 4}$

↓

Constant

Θ(1) Average case

Ω(1) Best case

O(1) → Constant Complexity

↑

BigOh

Ex

Time

```
Swap(a,b)
{   temp = a;  → 1 ms (stats)
       a = b;  → 1 ms
       b = temp; → 1 ms
}
```

$f(n) = 3$ ms

↑

constant

Space

(memory)

a → 1 unit

b → 1unit

temp → 1unit

$f(n) = 3$ units

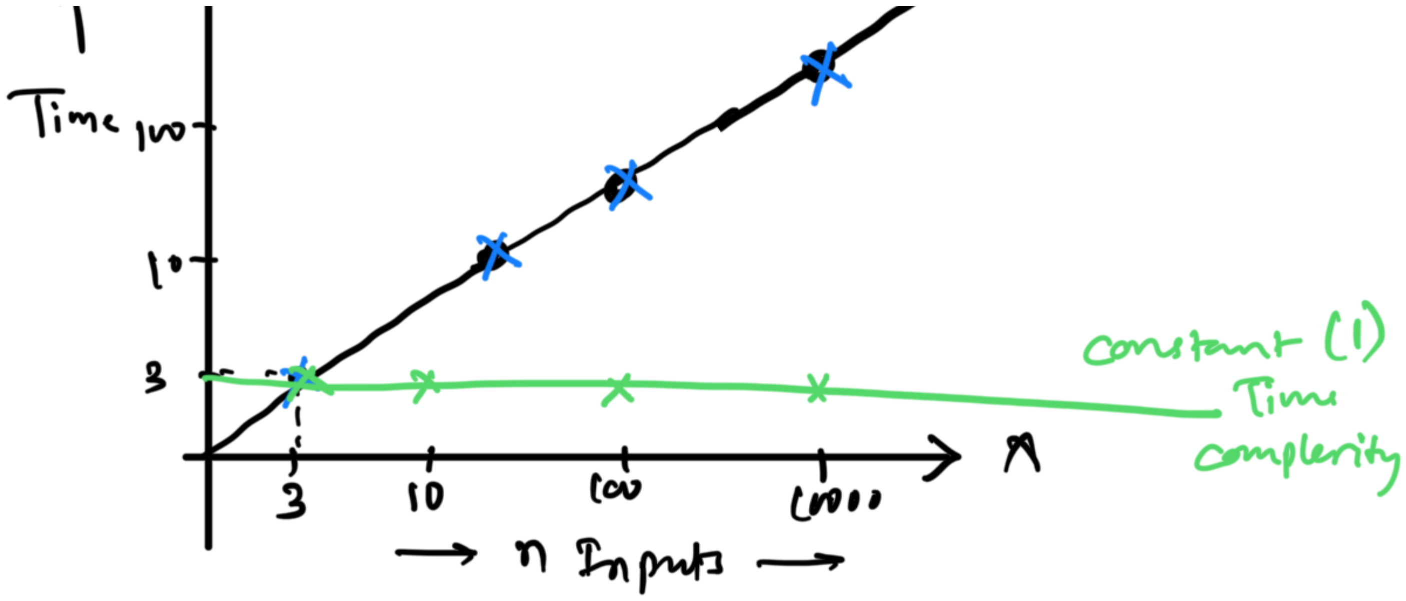
↑

Constant

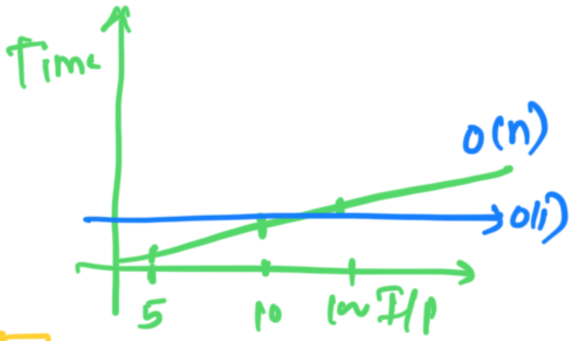
O(1)

Constant Time Comp O(1)

Linear (n) Time Complexity



3 sec ←	3 inputs →	3 ms	Rel^n bet^n i/p values and time taken
3 sec ←	10 inputs →	10 ms	
3 sec ←	100 inputs →	100 ms	
3 sec ←	10000 inputs →	10000 ms.	
Constant O(1)		Linear O(n)	



Optimization ⇒

Ex:2

i=0,1,2,3,4,5 ~~A~~ = [2 | 3 | 4 | 5 | 7] (0)

~~n~~ = 5

```
Algorithm: sum (A, n)                          Time      | Space
{                                                        |
    s=0;                              →  1               | A → n
    for(int i=0; i<n; i++)   → 5+1    →  n+1   (n+1)     | n → 1
          1 2 3 4 5                                      | i → 1
    {                                                    | S → 1
        S = S + A[i];        → 5      →  n     (n)       | f(n)=n+3
          0+2=2+3=5+4                                    |   ↓
          21 = 7+14 = 5+9"                               | O(n)
    }
    return S;                         →  1.
}
```

$f(n) = 2n + \cancel{3}$ → O(n)

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

$\rightarrow$ O(n) + $\cancel{O(1)}$

Consider.

1. Ignore constants
2. Consider Time Complexity

★ Important Rule

eq $f(n) = 2n^2 + 3n + 7$

O(n²) + O(n)

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

$= O(n^2)$

$n=3$

3x3

Ex 3 Algorithm

Time | Spau

$A = n \times n = 3 \times 3 = 3^2 \Rightarrow n^2$

m×n | m×n

```
Add ( A, B, n)
{
  for(i=0; i<n; i++)          — 1        n+1
  {
    for(j=0; j<n; j++)        — 2        n(n+1)
    {
      c[i,j] = A[i,j] + B[i,j];          n(n)
    }
  }
}
```

$A \rightarrow \overset{m \times n}{n^2}$

$B \rightarrow n^2$

$C \rightarrow n^2$

$n \rightarrow 1$

$i \rightarrow 1$

$j \rightarrow 1$

$f(n) = 3n^2 + \cancel{3}$

$O(n^2)$

$f(n) = n+1 + n(n+1) + n(n)$

Calcd

$= n+1 + n^2 + n + n^2$

$= \boxed{2n^2 + 2n + 1}$

$f(n) = 2n^2 + 2n + \cancel{1}$

$\boxed{= O(n^2)}$

square time complexity

$A = n \times n$

$n=3$

4

$4 \times 4 = 16$

n^2

3x3

n×n

n^2

A = m×n

A = m×n

B = m×n

C = m×n

i — 1

j — 1

n — 1

$f(n) = 3mn + \cancel{3}$

$= \cancel{3}(n \times n)$

$= O(n^2)$

Ex: 4

```
Mul (A, B, n)
{   for (i=0; i<n; i++)  ———————— n+1
    { for (j=0; j<n; j++)  ———————— n × (n+1)
      {   c[i,j]=0;  ———————— n × n
          for (k=0; k<n; k++)  ———————— n × n × (n+1)
          {
            c[i,j] = c[i,j] + A[i,k] + B[k,j];  — n × n × n
          }
      }
    }
}
```

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

$$= n+1 + n^2 + 1 + n^2 + n^3 + n^2 + n^3$$

$$= \cancel{2}n^3 + 3\cancel{n^2} + \cancel{2n} + \cancel{1}$$

$$= O(n^3) \longrightarrow \text{Time Complexity}$$

space

A → n^2

B → n^2

C → n^2

n → 1

i → 1

j → 1

k → 1

$$f(n) = \cancel{3}n^2 + \cancel{4}$$

$$O(n^2)$$

Remember:

```
for (i=0; [i<n]; [i++])  ———→ n+1
{                                      → O(n)
    stmt  ————————————————→ n
}
```

}

$O(n)$

Ex 2

```
for (i=0; i>n; i--)
{
        st;
}
```

$\longrightarrow O(n)$

Ex 3

~~2~~ ~~4~~ ~~6~~ ~~8~~ 10

10

```
for (i=~~0~~ 1; i<n; i=i+20)
{
        sop(i)
        ~~st~~;
}
```

$\longrightarrow \frac{n}{2}$

$O(\frac{n}{\cancel{2}})$

$O(n)$

0
2
4
6
8

5

```
for (i=0; i<n; i=i+2)
{
        st;
}
```

$\longrightarrow (n+1)$

$\longrightarrow \frac{n}{2}$

$f(n) = n+1+\frac{n}{\cancel{2}}$

$= \cancel{2}n+1$

$= O(n)$

Ex

```
for (i=0; i<n; i=i+20)
{
        st;
}
```

$\longrightarrow n+1$

$\longrightarrow \frac{n}{20}$

$f(n) = O(n)$