Time Complexity – Time, Space → Asymptotic

Asymptotic: Best, Avg, Worst — Ω, Θ, O

Ex 1

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

$f(n) = \boxed{3}$

$O(1)$

temp — 1
a — 1
b — 1

$S(n) = 3$

$O(1)$

Constant Time Complexity

Ex: 2

```
Sum(A,n)
{
    s = 0;                     —— 1
    for (i=0; i<n; i++)        → (n+1)
    {
        s = s + A[i];          → n
    }
    return s;                  —— 1
}
```

$f(n) = 2n+3$

$2n \rightarrow O(n)$, $3 \rightarrow O(1)$

$A \rightarrow n$
$n \rightarrow 1$
$s \rightarrow 1$
$i \rightarrow 1$

$S(n) = n+3$

$O(n)$

$O(n)$
Linear Time Complexity

Ex 3: Addition of Matrix

```
Add(A,B,n)
{
    for (i=0; i<n; i++)        → (n+1)
    {
        for (j=0; j<n; j++)    → n (n+1)
```

```
{
    C[i,j] = A[i,j] + B[i,j];   → n
                                → n
    }
  }
}
```

Time Complexity → $f(n) = (n+1) + n(n+1) + n(n)$

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

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

$\boxed{O(n^2)}$ → quadratic Time Complexity

Space Complexity →

$n=3$ $\quad$ 3x3

$A \to n \times n = n^2$

$B \to n \times n = n^2$

$C \to n \times n = n^2$

$n \to 1$

$i \to 1$

$j \to 1$

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

$\boxed{O(n^2)}$ → quadratic Time complexity

Ex 4 Multiply Matrix

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

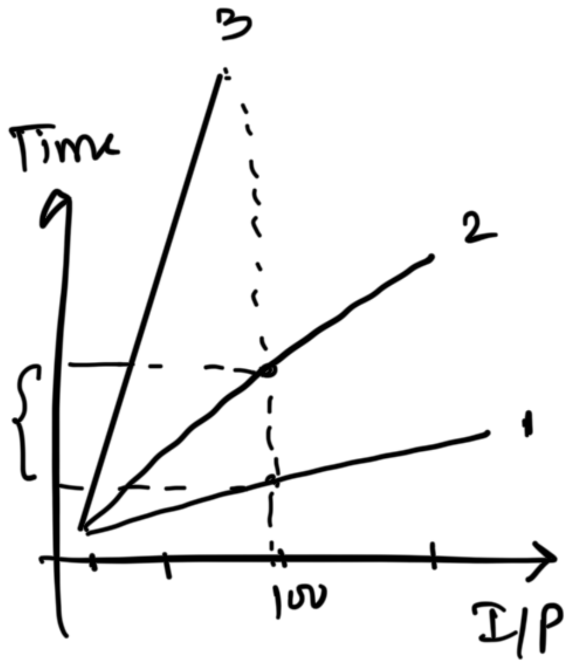
} }

$$f(n) = (n+1) + n(n+1) + n^2 + n^2(n+1) + n^3$$
$$= \underline{2n^3} + 3n^2 + 2n + 1$$
$$O(n^3) \longrightarrow \text{Cubic Time Complexity}$$

Space Complexity -

$A \rightarrow n \times n = n^2$
$B \rightarrow n^2$
$C \rightarrow n^2$
i — 1
j — 1
k — 1
n — 1

$$S(n) = 3n^2 + 4$$
$$O(n^2)$$



1 → linear
2 → Manageable
3 → Exponential time growth

$O(1)$, $O(n)$, $O(n^2)$, $O(n^3)$

$O(n^k)$ $O(2^k)$ $O(n!)$ $O(\log n)$

Ex:

```
for (i=0; i<n; i++)   ——→ n+1
{                     — n (10)  ——→ O(n)
}
```

(0, 10 written above $i=0$ and n)

$$\boxed{O(n)}$$

Ex for (i=n ; i>0 , i--) $\longrightarrow n+1$

{

—— $\longrightarrow n$

}

$\boxed{O(n)}$

Ex: for (i=1; i<n; i=i+2) $\longrightarrow \frac{n+1}{2}$

{

—— $\longrightarrow \frac{n}{2}$

}

$f(n) = \frac{n}{2} \Rightarrow \boxed{O(n)}$

Ex for (i=1; i<n; $\boxed{i=i+20}$)

{

st — ; $\longrightarrow \frac{n}{20}$

}

$f(n) = \frac{n}{20} \Rightarrow \boxed{O(n)}$

Ex for (i=0; i<n; i++) —— $n+1$

{ for (j=0; j<n; j++) —— $n(n+1)$

{

—— $\longrightarrow n \times (\frac{n}{2})$

}

}

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

Ex: for (i=0; i<n; i++) $\rightarrow n+1$

{

for (j=0; $\boxed{j<i}$; j++) $\rightarrow n(n+1)$

{

st;

} $\rightarrow n \times n$

}

i	j	Ex
0	0 ✗	0
1	0 ✓ 1 ✗	1
2	0 ✓ 1 ✓ 2 ✗	2
3	0 ✓ 1 ✓ 2 ✓	3

$3 \times$

$$\text{Time Ex} = \boxed{1+2+3+\dots n}$$

$$f(n) = \frac{n(n+1)}{2} \rightarrow \text{formula for sum of } n \text{ natural numbers}$$

$$\boxed{n} \rightarrow \boxed{n}$$

$$= \frac{n^2+n}{2}$$

$$O(n^2)$$

Ex

```
for (i=0; i<n; i++)   → (n+1)
{
    sop(i);           → n
}
```

Ex

```
for (i- ...)              ———— n+1
{
    for (j ...)           — n+1 ———— × n = n(n+1)
    {
    }                     — n ———— × n = n×n
}
```

$2 \quad 2^2 \quad 2^3$

Ex:

```
for (i=1; i<n; i=i*2)
{
    st ...
}
```

i
1
$1 \times 2 = 2$
$2 \times 2 = 2^2$
$2^2 \times 2 = 2^3$
$2^3 \times 2 = 2^4$
⋮
$2^k \dots n^{th}$ term

Induction proof → $i \geq n$; Assumption — ①

$\because \; i = 2^k$ — ②

$$2^k \geq n$$

$2^k = n \longrightarrow$ Apply log both the

$2^k = n$ Apply log both sides

$$k = \log_2 n$$

$$\boxed{O(\log_2 n)}$$

<u>Ex</u> for(i=n ; <u>i>=1</u> ; $\boxed{i=i/2}$)
{
st . . .
}

$\to i/2$

$\to n/2$

$\to \frac{n}{2} \times \frac{1}{2}$

$= \frac{n}{2^2}$

$\frac{n}{4} \times \frac{1}{2} = \frac{n}{2^3}$

$= \vdots$

$\frac{n}{2^k}$

Assume $i < 1$

$$\frac{n}{2^k} < 1$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

Apply log both sides

$$k = \log_2 n$$

$$\boxed{O(\log_2 n)}$$

Ex: 4 8 p=0;
for(i=1 ; $\boxed{p \le n}$; i++)
{
$\boxed{p = p + i;}$
}

i	p
1	0+1=1
2	1+2=3
3	1+2+3=6
4	1+2+3+4
⋮	
$n^{th} \to$ k	$\underbrace{1+2+3+4 \cdots + k}_{k(k+1)}$

Assume $p > n$

$$p = \frac{k(k+1)}{2}$$

$k^2 + k$

$$\frac{k^2+k}{2} = p > n \qquad \frac{\cdots}{2}$$

$$\frac{k^2+k}{2} > n$$

$$k^2 > n$$

$$k > \sqrt{n}$$

$$O(\sqrt{n})$$

Ex:

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

$$\underline{i*i} > \underline{n}$$

$$i^2 = n$$

$$i = \sqrt{n}$$

$$\boxed{O(\sqrt{n})}$$

Ex:

```
p=0;
for (i=1; i < n; i=i*2)
{   p++;
}
for ( j=1; j < p; j=j*2)
{
        st ..;
}
```

$p = \log n$

n

$\log p$

$O(\log \log n)$

Ex

$n+1 \leftarrow$ for (i=0; i<n; i++) $\rightarrow O(n)$

{

$n+1 \leftarrow$ for (j=0; j<n; $\boxed{j=j*2}$) $\rightarrow n \log n$

n {

{

st $\rightarrow \log n$ $\rightarrow n \boxed{\log n}$

}

}

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

$O(n \log n)$

$\log n - n - \boxed{n \log n} - n^2$

Summary →

for (i=0; i<n; i++) $\longrightarrow O(n)$

for (i=0; i<n; i=i+2) $\longrightarrow \frac{n}{2}$ $O(n)$

for (i=n; i>1; i--) $\longrightarrow O(n)$

for (i=1; i<n; i=i*2) $\rightarrow O(\log_2 n)$

for (i=1; i<n; i=i*3) $\rightarrow O(\log_3 n)$

Types of Function

(Best) $O(1) \rightarrow$ Constant $\rightarrow O(\log n) \rightarrow$ Logarithmic

$O(n) \rightarrow$ Linear

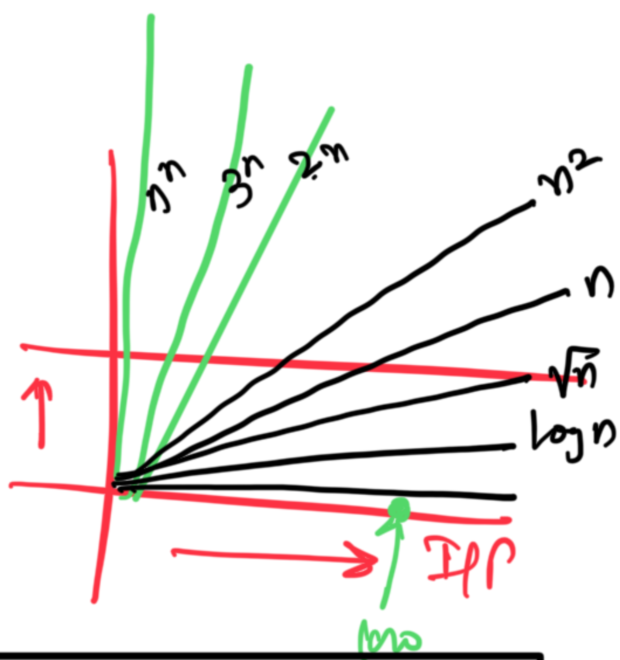
$O(n^2) \rightarrow$ Quadratic

$O(n^3) \rightarrow$ Cubic

worst $\rightarrow O(2^n) \rightarrow$ Exponential

$O(3^n) \rightarrow$

$O(n^n) \rightarrow$



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

$< 2^n < 3^n < \dots < n^n$

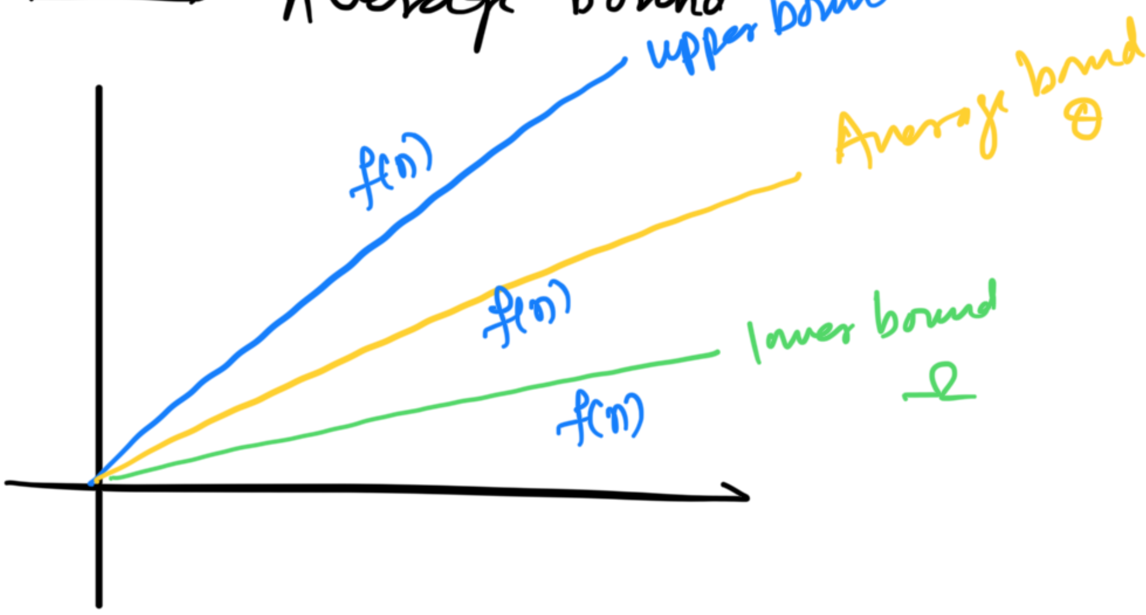
* Important *

Asymptotic Notations

$O \longrightarrow$ upper bound

$\Omega \longrightarrow$ lower bound

$\Theta \longrightarrow$ Average bound



$f(n) = O\,g(n) \longrightarrow$ Big-oh $\longrightarrow$ worst case

upper bound $\leftarrow$ $\boxed{f(n) \leq c * g(n)}$ $\forall\ n \geq n_0$ (1)

$f(n) = \Omega\,g(n) \longrightarrow$ Omega $\longrightarrow$ Best Case

lower bound $\leftarrow$ $\boxed{f(n) \geq c * g(n)}$ $\forall\ n \geq n_0$ (2)

$f(n) = \Theta\,g(n) \longrightarrow$ Theta $\longrightarrow$ Average Case

lower bound + Upper bound $\Rightarrow$ $\boxed{c_1 * g(n) \leq f(n) \leq c_2 * g(n)}$

$c_1, c_2,$ are constant & n_0

Ex:

Test (n)

Recurrence Relation

{

1 — if (n > 0)

1 — {

sop()

}

$\boxed{T(n-1)}$ — Test(n-1)

}

}

$f(n) = T(n) = T(n-1) + 2$

$=$

$f(n) = 2^n$

$O(2^n)$