$$\frac{k-1}{}$$

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

Space $\rightarrow O(n^2)$

Time Complexity

ex 1:

```
for(i=n ; i>0; i--)  — n+1
{
   //Statement        — n
}
```

$$\overline{\quad O(n) \quad}$$

ex 2:

```
for(i=0; i<n; i++){          — n+1
  for(j=0; j<n; j++){        — n x(n+1)
    // Staten                — n x n
}}
```

$$\overline{\quad O(n^2) \quad}$$

ex 3:

```
for(i=0; i<n ; i++){
   for(j=0; j<i; j++){
       ~~
   }
}
```

i	j	~
0	0x	0
1	0 1x	1
2	0 1 2x	2
3	0 1 2 3x	3
:		
n	0 : n	n

$$\frac{n(n+1)}{2}$$

$O(n^2)$

PAGE No.
DATE / /

ex4: $p=0$

```
for(i=1; p<=n; i++){
    p=p+i;
}
```

i	p
1	$0+1=1$
2	$1+2=3$
3	$1+2+3$
4	$1+2+3+4$
⋮	
k	$1+2+\cdots+k$

assume $p>n$

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

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

$k^2>n$

$k>\sqrt{n}$

$O(\sqrt{n})$

ex5:

```
for(i=1 ; i<n ; i=i*2){
    //---
}
```

i
1
$1\times2=2$
$2\times2=2^2$
$2^2\times2=2^3$
⋮
2^k

assm $i>=n$

$\because i=2^k$

$2^k>=n$

$k=\log_2 n$

$O(\log_2 n)$

ex6:

```
for(i=n ; i>=1 ; i=i/2){
    //---
}
```

i
n
$n/2$
$n/2^2$
$n/2^3$
⋮
$n/2^k$

asm $i<1$

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

$n=2^k$

$k=\log_2 n$

$O(\log_2 n)$

ex7:

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

i
0
1
2
3
⋮
K

assm $K^2<n$

$K<\sqrt{n}$

$O(\sqrt{n})$

ex 8. for(i=0; i<n; i++)

{ — } ——— n

for (j=0; j<n; j++)

{ — } ——— n

———

2n

O(n)

ex 9. p=0

for(i=1; i<n; i=i*2)

{ p++; } ——— $\log_2 n$

for(j=1; j<p; j=j*2)

{ ~~//—~~ ; } ——— $\log_2 p$

———

O(log log n)

i	p	j
1	1	1
2	2	2
2^2	3	2^2
2^3	.	2^3
⁝	⁝	⁝
2^k	k	2^m

assume $2^k < n$

$k * \log_2 n$

$O(\log_2 n)$

assume $2^m < p = k$

$2^m < p$

~~$k = \log_2 m$~~

$m = \log_2 p$

$O(\log_2 p)$

ex 10. for (i=0; i<n; i++) ——— n

{ for(j=1; j<n; j=j*2) ——— $n \times \log_2 n$

{ { — ——— $n \times \log_2 n$

}

}

———

$2n \log n + n$

$O(n \log n)$

Summary

for (i=0; i<n; i++) ——— O(n)

for(i=0; i<n; i=i+2) ——— O(n)

for(i=n; i>1; i--) ——— O(n)

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

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

for (i=n; i>1; i=i/2) ——— $O(\log_2 n)$

DATE / /

```
i=0            ——— 1
while(i<n){    ——— n+1
  //stm        ——— n
  i++          ——— n
}
```

$f(n) = 3n+2$

$O(n)$

ex1:

```
a=1;          ——— 1
while(a<b)    ———
{
  stm //
  a=a*2;
}
```

a
1
1×2=2
$2\times2=2^2$
$2^2\times2=2^3$
⋮
2^k

assum

terminat $a \geq b$

$a = 2^k$

$2^k \geq b$

$2^k = b$

$k = \log_2 b$

$O(\log_2 n)$

ex2:

```
Algorithm Test (n)
{
    if (n<5)
    {
      printf("%d",n)      ——— 1
    }
    else{
     for(i=0;i<n;i++)
     { printf("%d",i); ——— n
     }
    }
}
```

best case time — O(1)

worst case time — O(n)

PAGE No.
DATE / /

Types of Time Functions

$O(1)$ — constant $f(n) = 2, 5, 5000$, etc

$O(\log n)$ — logarithmic

$O(n)$ — linear $f(n) = 2n+3$, $500n + 700$, etc

$O(n^2)$ — quadratic

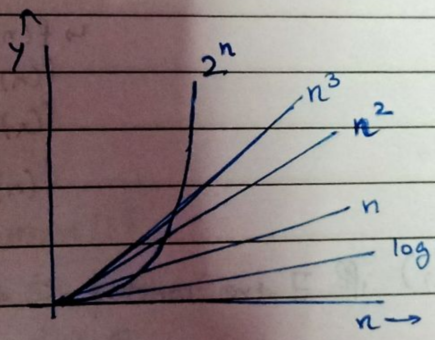
$O(n^3)$ — cubic

$O(2^n)$ — Exponential

$O(3^n)$

$O(4^n)$

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



Asymptotic Notations

O big-oh (upper bound)

Ω big-omega (lower bound)

θ theta (Average bound)

Big - Oh

The function $f(n) = O(g(n))$ iff $\exists$ +ve constant c & n_0

such that $f(n) \leq c * g(n) \ \forall \ n \geq n_0$

ex: $f(n) = 2n+3$

$2n+3 \leq 10 * n \quad \forall \ n \geq 1$

$\therefore f(n) = O(n)$

$\underbrace{1 < \log n < \sqrt{n}}_{\text{lower bound}} < \underbrace{n}_{\text{average bound}} < \underbrace{n \log n < n^2 < n^3 \cdots < 2^n < 3^n \cdots < n^n}_{\text{upper bound}}$

$\therefore f(n) = O(n)$ ✓

$f(n) = O(n^2)$ ✓

$f(n) = O(2^n)$ ✓

$f(n) = O(\log n)$ ✗

Omega

The function $f(n) = \Omega(g(n))$ iff $\exists$ +ve constant c & n_0

such that $f(n) \geq c * g(n) \ \forall \ n \geq n_0$

ex: $f(n) = 2n+3$

$2n+3 \geq 1 * \log n \quad \forall \ n \geq 1$

$f(n) = \Omega(n)$ ✓

$f(n) = \Omega(\log n)$ ✓

✗ $f(n) = \Omega(n^2)$

Theta

The function $f(n) = \Theta(g(n))$ iff $\exists$ +ve constant c_1, c_2 & n_0

such that $c_1 * g(n) \leq f(n) \leq c_2 * g(n)$

ex: $f(n) = 2n+3$

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

$f(n) = \Theta(n)$ ✓

$f(n) = \Theta(n^2)$