Design and Analysis of Algorithm

* Chapter Two

- Asymptotic Notations:-

✓ How to analyze an Algorithm

Example:-

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

$f(n) = 3$

$\Theta(1)$

* Each statement has one unit of time
* Each variable has one unit of time

Space Complexity
a → 1
b → 1
temp → 1

$s(n) = 3$

$\Theta(1)$

Example:-

Algorithm sum (A, n)
{
S = 0; → 1
For (i = 0; i<n; i++) → n+1
{
S = S + A[i]; → n
}
return S; → 1
}

$i=0$, $i=1$, $i=2$, $i=3$, $i=4$, $i=5$ ×

A = | 8 | 3 | 9 | 7 | 2 |, indices 0 1 2 3 4, n = 5

{Frequency count method}

$f(n) = 2n+3$

$= \Theta(n)$

Space
A = n
n = 1
S = 1
i = 1

$S(n) = n+3 = \Theta(n)$

Example:-

Algorithm Add (A, B, n) → sum of two matrices
{
For (i = 0; i<n; i++) → n+1
{ For (j = 0; j = n; j++) → n × (n+1) = n^2+n
{
C(i,j) = A[i,j] + B[i,j] → n × n = n^2
}
}
}

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

$= \Theta(n^2)$

Space
$A = n^2$
$B = n^2$
$C = n^2$
n = 1
i = 1
j = 1
k = 1

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

Exercise

Algorithm multiply (A, B, n)

{

$n+1$ — for (i = 0; i<n; i++)

$(n+1)\,n$ — { For (j = 0; j<n; j++

{

$n \times n$ — c[i, j] = 0;

$(n+1)\,n \times n$ — for (k = 0; k<n; k++)

{

$n \times n \times n$ — c[i, j] = c[i, j] + A[i, k] * B[k, j];

}

}

}

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

$= \Theta(n^3)$

Space

A — n^2

B — n^2

C — n^2

n — 1

i — 1

j — 1

k — 1

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

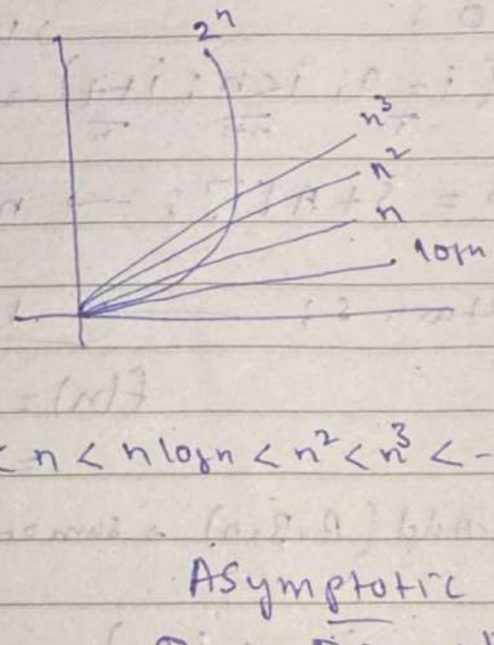
$O(n^2)$

Types of Time functions :-

$f(n) = 3$, $f(n) = 30$, $f(n) = 4000$ } $O(1)$

Classes of Functions:

* $O(1)$ → Constant
* $O(\log n)$ → Logrithemic
* $O(n)$ → Liner
* $O(n^2)$ → Quadritic
* $O(n^3)$ → Cubic
* $O(2^n)$ → Exponential



$\log_2 1 = 0$

$\log_2 2 = 1$

$\log_2 4 = 2$

$n^{100} < 2^n$

$n^k < 2^n$

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

Compare class of function

Logn	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 → Big omega → lower bound

Θ → Theta → Average bound

$\sqrt{n}$

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

Lower bound — average bound — upper bound

Properties of Asymptotic Notation:-

* General properties:-

if $f(n)$ is $O(g(n))$

then $a * f(n)$ is $O(g(n))$

Example:-

$f(n) = 2n^2 + 5$ is $O(n^2)$

then $a=7$ $\quad 7(2n^2+5)$

$= 14n^2 + 35$ is also $O(n^2)$

* Reflexive properties:-

if $f(n)$ = is given then $f(n)$ is $O(f(n))$

Example:-

$f(n) = n^2$

$O(n^2)$

* Transitive properties.

if $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$ then $f(n) = O(h(n))$

Example:

$f(n) = n \quad g(n) = n^2 \quad h(n) = n^3$

then $f(n)$ is $O(n^2)$ and $g(n) = O(n^3)$

so $f(n) = O(n^3)$

* Symmetric properties-

if $f(n)$ is $\theta(g(n))$ then $g(n)$ is $\theta(f(n))$

Example:- $f(n) = n^2 \quad g(n) = n^2$

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

$g(n) = \theta(n^2)$