PAGE NO.
DATE: / /

# DAA (Design & Analysis of Algo.)

## Tutorial - 1

Name : Sakshi Sahu

Class Roll no. : 03

University Roll no. : 2017544

Date : 8/03/22

Section : CST SPL-1

Semester : 4th

Sakshi Sahu

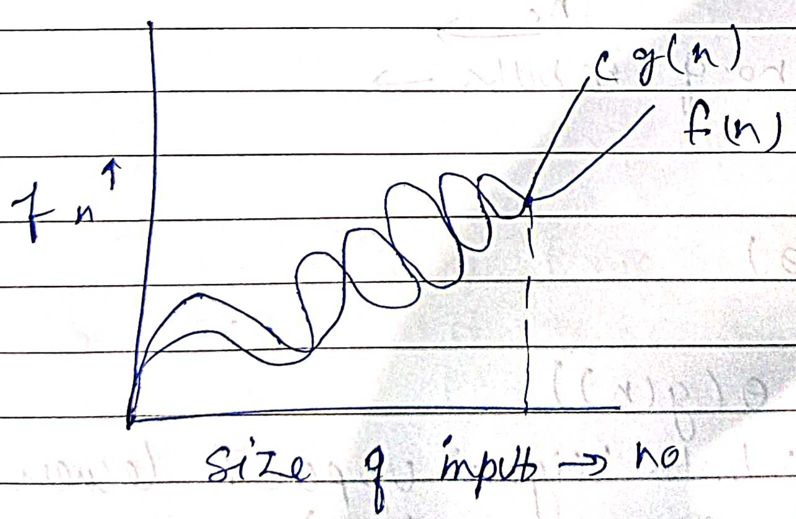
PAGE NO.
DATE: / /

Ans: Asymptotic Notation : The Notation used to describe the asymptotic running time of an algorithm are defined in terms of function whoes domain are the set of natural number.

1) Big O (O)

$f(n) = O(g(n))$
iff $f(n) \leq c \cdot g(n)$
$\forall \; n \geq n_0$
for some constant $c > 0$

$\Rightarrow$ g(n) is tight upper bound of f(n)



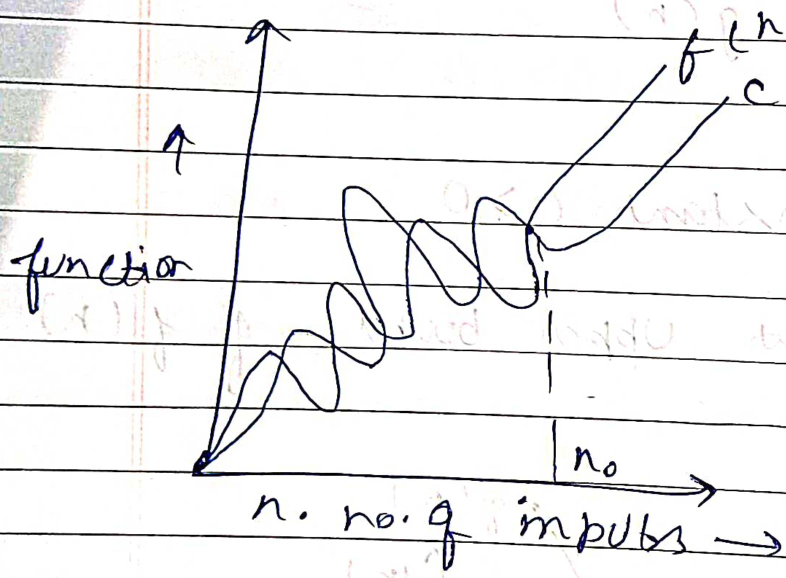

Sakshi Sahu

2) Big omega ($\Omega$)

$f(n) = \Omega(g(n))$
g(n) is 'tight' lower bound
of f(n)

$f(n) = \Omega(g(n))$
iff $f(n) \geq c \cdot g(n)$

$\forall \; n \geq n_0$ for some constant $c > 0$




3) Theta ($\Theta$)

$f(n) = \Theta(g(n))$
g(n) is both 'tight' upper & lower
bound of fn f(n)

$f(n) = \Theta(g(n))$

Sakshi Saw

PAGE NO.
DATE: / /

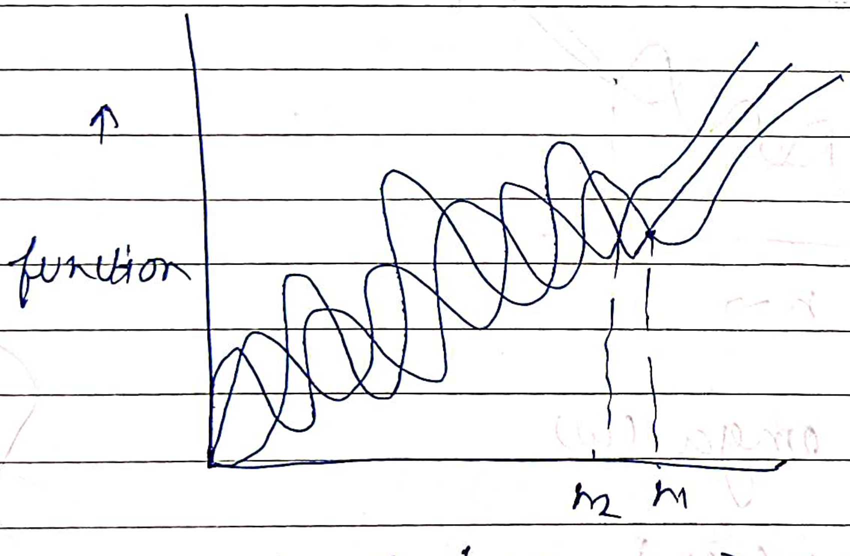
iff,

$c_1 g(n) \leq f(n) \leq c_2 \cdot g(n)$

$\forall n \geq \max(n_1, n_2)$
for some constants $c_1 > 0$ & $c_2 > 0$

c) ~~Small o (O)~~

$f(n) = o(g(n))$
g(n) is upper bound of fn




Sayshi Sahu

PAGE NO.
DATE: / /

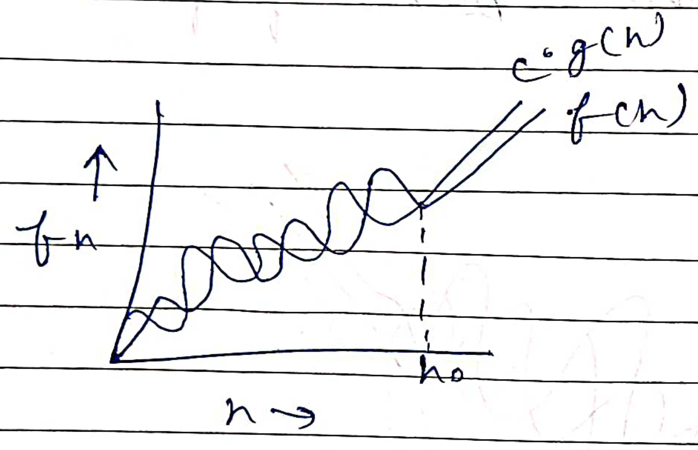
4) Small o (O)

$f(n) = o(g(n))$

g(n) is upper bound of f(n) f(n)

$f(n) = o(g(n))$

when $f(n) < c \cdot g(n)$
$\forall\ n > n_0$
& $\forall\ c > 0$



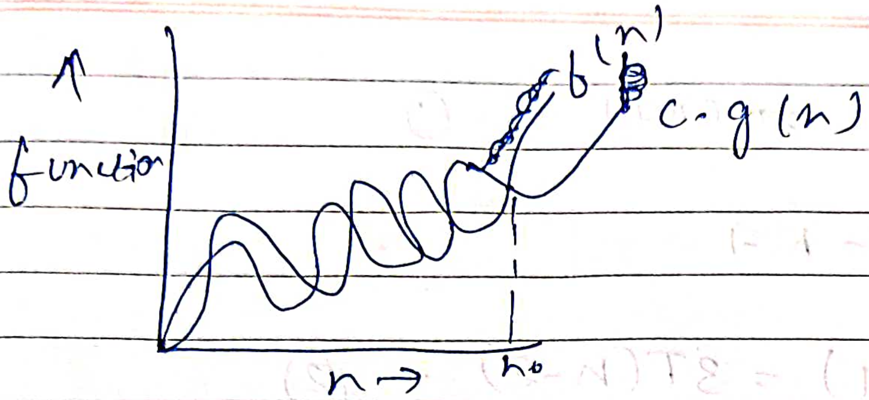

5) small omega (ω)

$f(n) = \omega(g(n))$
g(n) is lower bound of n f(n)
$f(n) = \omega(g(n))$

when $f(n) > c \cdot g(n)$
$\forall\ n > n_0$
& $\forall\ c > 0$

Sayeshi Saha

PAGE NO.
DATE: / /




**Ans 2** for (i = 1 to n) { i = 1 * 2 }

for (i = 1 to n) // i = 1 2 4, 8, --- n
{ i = i* } // O(1)

$\Rightarrow \sum_{i=1}^{n} 1+2+4+8+ --- n$

GP $k^{th}$ value $\Rightarrow T_k = ar^{k-1}$
$= 2^{k-1} \times 1$

$n = 2^{k-1}$

$2n = 2^k$

$\log 2n = k \log 2$

$\log_2 + \log n = k \log n$

$\log n+1 = k$

$\Rightarrow O(k) = O(1+\log n)$

$O(\log n)$ **Ans**

Sakshi Sahu

3)
An

$T(n) = 3T(n-1)$ --- ①

Put $n = n-1$

$T(n-1) = 3T(n-2)$ --- ②

from ① & ②

$T(n) = 3(3T(n-2))$
$= 9T(n-2)$ --- ③

Putting $n = n-2$ in ①

$T(n) = 3(T(n-3))$ --- ④
$T(n) = 27(T(n-3))$

$T(n) = 3^k (T(n-k))$

Putting $n-k = 0$
$n = k$

$T(n) = 3^n [T(n-n)]$
$T(n) = 3^n T(0)$
$T(n) = 3^n \times 1$ $[T(0) = 1]$
$T(n) = O(3^n)$

$T(n) = O(3^n)$

Ans

Sayasri Sahu

PAGE NO.
DATE: / /

4) $T(n) = \{ 2T(n-1) - 1$ if $n > 0$, otherwise 1 ;

Ans

$T(n) = 2T(n-1) - 1$ —— ①

let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ —— ②

from ① & ②

$T(n) = 2[2T(n-2) - 1] - 1$

$T(n) = 4T(n-2) - 2 - 1$ —— ③

let $n = n-2$

$T(n-2) = 2T(n-3) - 1.$ —— ④

from ③ & ④

$T(n) = 4[2T(n-3) - 1] - 2 - 1$

$T(n) = 8T(n-3) - 4 - 2 - 1$

$T(n) = 2^k T(n-k) . 2^{k+} - 2^{k+2} + ----$

$GP = 2^{k-1} + 2^{k-2} + 2^{k-3} + ----$

$a = 2^{k-1}$

$r = 1/2$

Sakshi Sahu

PAGE NO.
DATE: / /

$$= \frac{a(1-r^n)}{1-r}$$

$$= \frac{2^{k-1}(1-(1/2)^n)}{1/2}$$

$$= 2^k(1-(1/2)^k)$$

$$= 2^k - 1$$

let $n-k = 0$
$n = k$

$T(n) = 2^n T(n-n) - (2^n - 1)$
$T(n) = 2^n \cdot 1 - (2^n - 1)$
$T(n) = 2^n - (2^n - 1)$

$T(n) = O(1)$ Ans

Sakshi Sahu

PAGE NO.
DATE: / /

5 Ans- Let loop executes k times
loop will executes till $s \leq n$
After 1st

$S = S+1$

After 2nd
$S = S+1+2$

After 3rd
$S = S+1+2+3$

$S = S+1+2+ \dots k$ {Since the loop goes k times}

$\frac{k*(k+1)}{2} <= n$

$O(k^2) <= n$

$k = O(\text{root}(n))$ Ans

Sakshi sahu

6)
Ans

as $i^2 <= n$
$i <= \sqrt{n}$

$i = 1, 2, 3, 4, \dots \sqrt{n}$

$\sum_{i=1}^{n} 1+2+3+4+ \dots + \sqrt{n}$

$T(n) = \frac{\sqrt{n} \times (\sqrt{n}+1)}{2}$

$T(n) = \frac{n\sqrt{n}}{2}$

$T(n) = O(n)$ Ans

(7) for $k = k*2$
Ans

$k = 1, 2, 4, 8, \dots n$

GP $= 1, 2, 4, 8, \dots n$

$a = 1$
$r = 2$

Sum of n terms $= \frac{a(r^n - 1)}{r - 1}$

Sakshi Sahu

PAGE NO.
DATE: / /

$$n = \frac{1(2^k - 1)}{2-1} = \frac{1(2^k - 1)}{1}$$

$$n = 2^k - 1$$

On taking log both side,

$$\log_2 n = k \log_2 2 - \log_2 1$$

$$\log_2 n = k$$

| i | j | k |
|---|---|---|
| 1 | log n | log n * log n |
| 2 | log n | ⋮ |
| ⋮ | ⋮ | ⋮ |
| n | n | log n * log n |

$O(\log n * \log n) \Rightarrow O(n \log^2 n)$

Sakshi Sahu

PAGE NO.
DATE: / /

Ans 8

```
function (int n)
{
  if (n==1)
    return;          // O(1)
for (i = 1 to n) // i = 1,2,3,4 ---n => O(n)
{
  for (j =1 to n) // j = 1,2,3,4 ---n => O(n^2)
  {
    printf ("*")
  }
}
  function(n-3);     // T(n/3)
}
```

Using master's method

$T(n) = T(n/3) + n^2$

$a = 1 \quad b = 3 \quad f(n) = n^2$

$c = \log_3 1 = 0$

$n^c = 1 > f(n)$

$T(n) = O(n^2)$ Ans

Sakshi Sahu

PAGE NO.
DATE: / /

Ans 9

```
void function (int n)
{ for (i = 1 to n)          // O(n)
  { for (j=1 ; j<=n ; j+=i )
      printf ("*");
  }
}
```

for $i = 1 \rightarrow j = 1, 2, 3, 4, \ldots n$
$= n$

$i = 2 \rightarrow j = 1, 3, 5, \ldots n$
$= n/2$

$i = 3 \rightarrow j = 1, 4, 7, \ldots n$
$= n/3$

$i = n \rightarrow j = 1, \ldots n$
$= n/n$

$\Rightarrow \sum_{j=n}^{1} n + \frac{n}{2} + \frac{n}{3} + \ldots + 1$

Sakshi Sahu

PAGE NO.
DATE: / /

$$\sum_{j=n}^{1} n\left[1+\frac{1}{2}+\frac{1}{3}+\cdots\cdots\frac{1}{n}\right]$$

$$\sum_{j=n}^{1} n[\log n]$$

$T(n) = O(n \log n)$ Ans

Ans 10 Given: $a^k$ and $c^n$

and:

~~$n^k$ and $c^n$~~

$n^k = O(c^n)$

$n^k \leq ac^n$

$\forall$ $n \geq n_0$ and some constant $a > 0$

for $n_0 = 1$
$c = 2$

$1^k \leq a2^1$

$n_0 = 1$ and $c = 2$ Ans

Sakshi Sahu