TANISHQ
CST
34

DAA

Tutorials 1

Date / /

1. What do you understand by Asymptotic notation. Define different asymptotic notation with examples

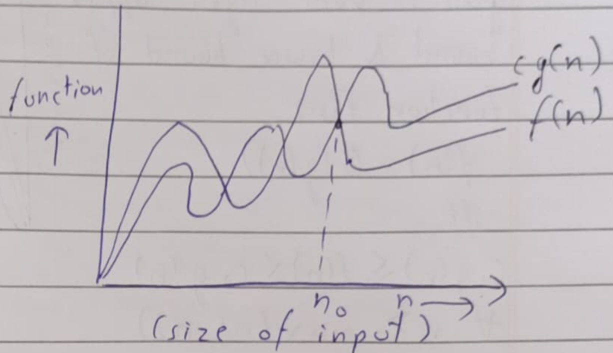
i) **Big O(n)**

$f(n) = O(g(n))$

iff

$f(n) \leq c\, g(n)$

$\forall\ n \geq n_0$



for some constant, $c > 0$

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

Ex → $f(n) = n^2 + n$

$g(n) = n^3$

$n^2 + n \leq c \cdot n^3$

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

ii) **Big Omega (Ω)**

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

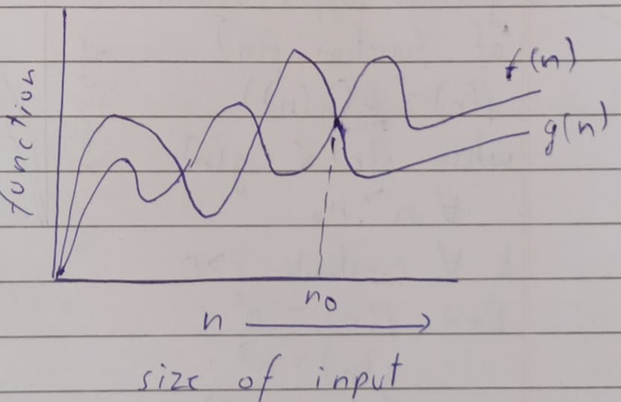
g(n) is "tight" lower bound of function f(n)

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

iff

$f(n) \geq c\, g(n)$

$\forall\ n \geq n_0$



for some constant $c > 0$

Ex -

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

$g(n) = n^2$

$n^3 + 4n^2 = \Omega(n^2)$

Tanishq

Date / /

iii) Big Theta (Θ)

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

g(n) is both "tight" upper bound & lower bound of function f(n)

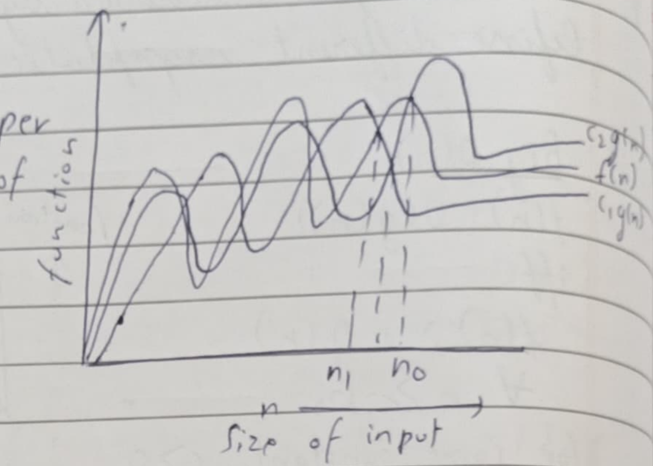
$f(n) = \Theta(g(n))$

iff

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

$\forall\ n \geq \max(n1, n2)$

for some constant $c1 > 0$ & $c2 > 0$



Ex-

$3n+2 = \Theta(n)$ as $3n+2 \geq 3n$ &

$3n+2 \leq 4n$ for n, $K1 = 3$, $K2 = 4$, & $n_0 = 2$

iv) Small O (o)

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

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

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

when $f(n) < c\,g(n)$

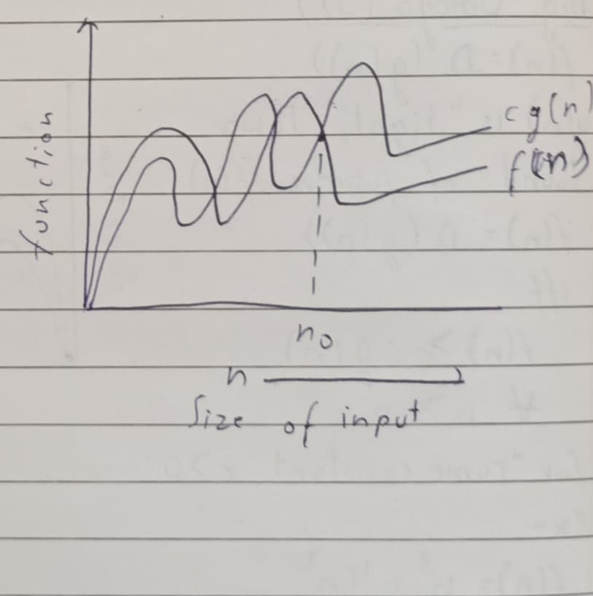
$\forall\ n > n_0$

& $\forall$ constants, $c > 0$

Ex→ $f(n) = n^2$

$g(n) = n^3$

$n^2 = o(n^3)$



v) Small Omega (ω)

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

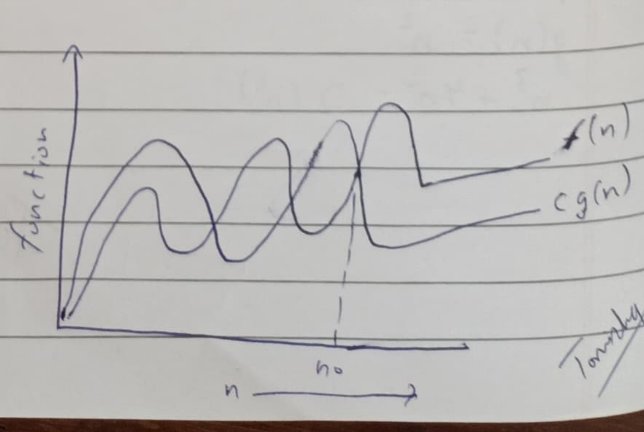
g(n) is lower bound of func^n f(n)

$f(n) = \omega(g(n))$ when

$f(n) > c \cdot g(n)$

$\forall\ n > n_0$

& $\forall$ constants, $c > 0$



Date / /

2. What should be time complexity of -
for (i=1 to n) { i = i * 2; }

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

$$i = \overbrace{1, 2, 4, \ldots, n}^{K}$$

$a = 1$, $r = \frac{b_2}{b_1} = 2$

G.P K^{th} value, $t_K = ar^{K-1}$

$t_K = 2^{K-1}$

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

$2n = 2^K$

$\log_2(2n) = K \log_2 2$ $\{\log_a a = 1\}$

$K = \log_2 2n$

$K = \log_2 2 + \log_2 n$ $\{\log ab = \log a + \log b\}$

$K = 1 + \log_2 n$

$O(1 + \log_2 n)$

$\Rightarrow \underline{\underline{O(\log n)}}$

Tanishq

Date / /

3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

Ans

$T(n) = 3T(n-1)$ —— ①

put $n = n-1$ in ①

$T(n-1) = 3T(n-2)$ —— ②

put value of $T(n-1)$ from ② to ①

$T(n) = 3[3T(n-2)] \Rightarrow 9T(n-2)$ —— ③

put $n = n-2$ in ①

$T(n) = 3T(n-3)$ —— ④

put value of $T(n-2)$ from ④ to ③

$T(n) = 9[3T(n-3)]$

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

By generalizing, $T(n) = 3^k T(n-k)$ —— ⑤

Let $n-k = 1$

$k = n-1$

put value of k in ⑤

$T(n) = 3^{n-1} T(n-n+1)$

$T(n) = 3^{n-1} T(1)$

$T(n) = \frac{3^n}{3} \times 1$

~~T(n)~~

$O(3^n)$

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

Ans.

$T(n) = 2T(n-1)-1$ —— ①

put $n=n-1$ in ①

$T(n-1) = 2T(n-2)-1$ —— ②

put value of $T(n-1)$ from ② to ①

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

$T(n) = 4T(n-2)-2-1$ —— ③

put $n = n-2$ in ①

$T(n-2) = 2T(n-3)-1$ —— ④

Tanishq

put value of $T(n-2)$ from (4) to (3)

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

$$T(n) = 8T(n-3) - 4 - 2 - 1 \quad \text{—— (5)}$$

$$T(n) = 8T(n-3) - 7$$

By generalizing, we get

$$T(n) = 2^k T(n-k) - (2^k - 1) \quad \text{—— (6)}$$

Let $n - k = 1$

$k = n - 1$

put k in (6)

$$T(n) = 2^{n-1} T(n-n+1) - (2^{n-1} - 1)$$

$$= 2^{n-1} T(1) - 2^{n-1} - 1$$

$$= 2^{n-1} - 2^{n-1} + 1$$

$$= 1$$

$$\Rightarrow \underline{\underline{O(1)}}$$

5. What should be time complexity of

```
int i=1, s=1;
while (s<=n)
{ i++; s=s+i;
  printf("#");
}
```

$\Rightarrow$ $i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad - - - - - -$

$s = 1 + 3 + 6 + 10 + 15 + - - - - - -$

sum of $s = 1 + 3 + 6 + 10 + - - - - + n$ —— (1)

Also $s = 1 + 3 + 6 + 10 + - - - - - + T_{n-1} + T_n$ —— (2)

$$0 = 1 + 2 + 3 + 4 + - - - - + n - T_n$$

$$T_k = 1 + 2 + 3 + 4 + - - - + k$$

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

Tanishq

Date / /

$\Rightarrow$ for K iterations

$1+2+3+ \cdots + K <= n$

$$\Rightarrow \frac{K(K+1)}{2} <= n$$

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

$$O(K^2) <= n$$

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

$$\underline{\underline{T(n) = O(\sqrt{n})}}$$

6. Time complexity of

```
void fn(int n)
{ int i, count=0;
  for(i=1; i*i<=n; ++i)
}
```

$\Rightarrow$ As $i^2 <= n$

$i <= \sqrt{n}$

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

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

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

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

$$\underline{\underline{T(n) = O(n)}}$$

Tanishq

Date / /

7\. Time complexity of

```
void fn(int n)
{
    int i, j, k, count=0;
    for(i=n/2; i<=n; ++i)
        for(i=1; j<=n; j=j*2)
            for(K=1; K<=n; K=K*2)
                cout++;
}
```

$\Rightarrow$ For $K = k^2$

$K = 1, 2, 4, 8, ----, n$

G.P $\Rightarrow$ $a = 1$, $r = 2$

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

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

$n \Rightarrow 2^K - 1$

$\log n \Rightarrow K$

i	j	K
1	log n	log n * log n
2	log n	log n * log n
⋮	⋮	⋮
n	log n	log n * log n

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

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

Tanishq

Date / /

8. Time complexity of-

```
function (int n)
{ if (n==1) return;
  for (i=1 to n)
    for (j=1 to n)
      printf("*");
    }
  }
  function (n-3);
}
```

Ans. For :- for (i=1 to n)

we get j=n times every turn

$\therefore i \times j = n^2$

Now, $T(n) = n^2 + T(n-3);$

$T(n-3) = (n^2 3)^2 + T(n-6);$

$T(n-6) = (n^3 6)^2 + T(n-5);$

K Times

⋮

$T(1) = 1;$

Now subs. each value in T(n)

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \cdots + 1$

Let

$n - 3K = 1$

$K = (n-1)/3$

Total terms $= K+1$

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \cdots + 1$

$T(n) \approx Kn^2$

$T(n) \approx \frac{(n-1)}{3} \times n^2$

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

Tanishq

Date / /

9. Time complexity of -

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

Ans: for :- $i=1$ $\quad j = 1+2+\cdots \quad (n \geq j+i)$

$i=2 \quad j = 1+3+5+\cdots$ "

$i=3 \quad j = 1+4+7\cdots$ "

⋮

m^{th} term of A.P is

$T(m) = a + d \times m$

$T(m) = 1 + d \times m$

$(n-1)/d = m$

for $i=1$ $\quad (n-1)/1$ times

$i=2 \quad (n-1)\,2$ times

$i=3 \quad (n-1)/3$ times

⋮

$i=n-1 \quad 1$

we get

$T(n) = i_1 j_1 + i_2 j_2 \cdots i_{n-1} j_{n-1}$

$= \frac{(n-1)}{1} + \frac{(n-2)}{2} + \frac{(n-3)}{3} + \cdots + 1$

$= n + \frac{n}{2} + \frac{n}{3} + \cdots + \frac{n}{n-1} - n \times 1 +$

$= n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{n-1}\right] - n + 1$

$= n \times \log n - n + 1$

Since $\int \frac{1}{x} = \log x$

$\underline{T(n) = O(n \log n)}$

Tanishq

Date / /

10. For the functions, n^k & c^n, what is the asymptotic relationship b/w these functions? Assume that $k >= 1$ & $c > 1$ are constants. Find out the value of c & no for which relation holds

Ans. As given n^k & c^n

relation between n^k & c^n is

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

$$n^k \leq a(c^n)$$

$\forall\, n \geq n_0$ &

constant, $a > 0$

for $n_0 = 1$

$c = 2$

$\Rightarrow 1^k \leq a^{2 \cdot 1}$

$\Rightarrow n_0 = 1$ & $c = 2$

Tanishq