Sakshi Semwal
38 - CST SPL1

DAA - Tutorial 1

Page

Topic

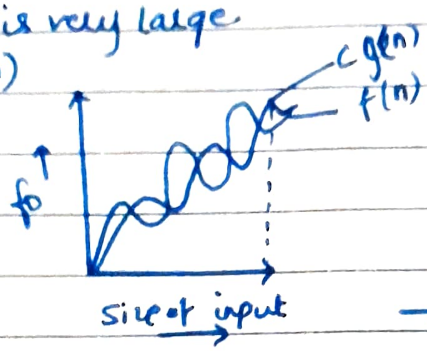
1. Asymptotic Notation

↓

(tending to infinity)

They help you find the complexity of an algorithm when input is very large.

1) Big O(O)



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

iff $f(n) \leq c \cdot g(n)$

$\forall\ n \geq n_0$

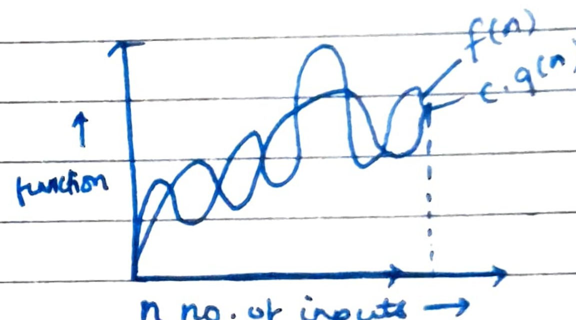
for some constant $c>0$

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

2) Big Omega (Ω)

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

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



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

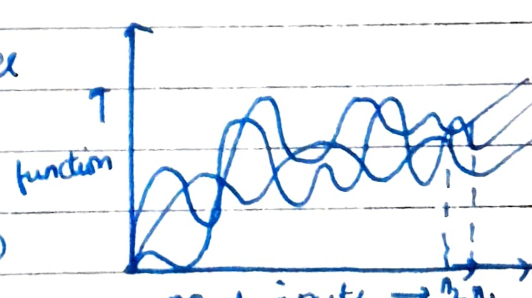
iff $f(n) \geq c(g(n))$

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

3) Theta (Θ)

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

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



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

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

$\forall\ n \geq \max(n_1, n_2)$

for some constant $c_1>0$ & $c_2>0$

Signature.................................

Date :
Topic :
Page :

4) Small o (o)

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

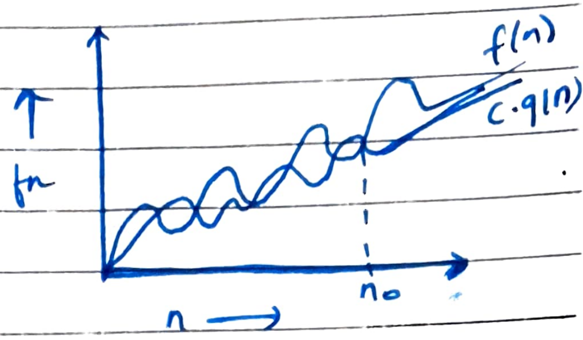
$g(n)$ is upper bound of fn. $f(n)$

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

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

$\forall\ n > n_0$

$\&\ \forall\ c > 0$



5) Small omega (ω)

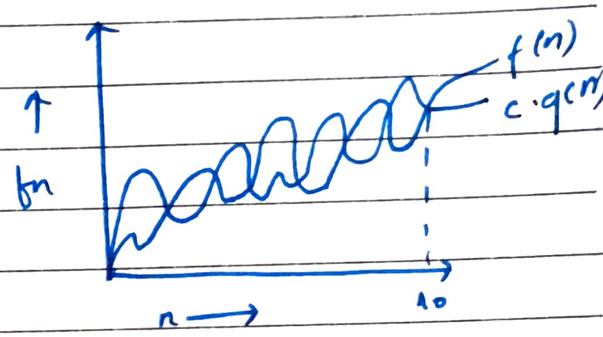
$f(n) = \omega(g(n))$

$g(n)$ is lower bound of fn $f(n)$

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

when $f(n) > c.g(n)$

$\forall\ n > n_0$ and $c > 0$



Q2. what should be time complexity of for(i=1 to n){i=i*2}

→ for(i = 1 to n) // $i = 1, 2, 4, 8 \ldots n$

{ i=i*2 } // $O(1)$

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

G.P k^{th} value → $T_k = ar^{k-1}$

$\rightarrow 1 \times 2^{k-1}$

$n = 2^{k}$

$2n = 2^{k}$

$\log 2n = k \log 2$

$\log_2 + \log n = k \log 2$

$\log n + 1 = k$

Signature.................................

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

$= O(\log n)$

STUDENT STYLE

Date : Page :
Topic :

Q3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

$T(n) = 3T(n-1)$ — ①

put $n = n-1$

$T(n-1) = 3T(n-2)$ — ②

from ① and ②,

$\rightarrow T(n) = 3(3T(n-2))$

$= 3T(n-2)$ — ③

putting $n = n-2$ in ①,

$T(n) = 3T(n-3)$ — ④

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

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

putting $n-k = 0$

$\rightarrow n = k$

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

$T(n) = 3^n T(0)$

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

$\underline{T(n) = O(3^n)}$

Q4. $T(n) = \{2T(n-1)-1$ if $n>0$, otherwise $1\}$

$T(n) = 2T(n-1) - 1$ — ①

let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — ②

from ① and ②, $T(n) = 2[2T(n-2)-1]-1$

$T(n) = 4T(n-2) - 2 - 1$ — ③

let $n = n-2$

$\Rightarrow T(n-2) = 2T(n-3) - 1$ — ④

Signature...............................

STUDENT STYLE

Date : ______ Page : ______
Topic : ______

from (3) and (4),

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

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

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

$\rightarrow$ G.P $= 2^{k-1} + 2^{k-1} + 2^{k-3}$

$a = 2^{k-1}$

$r = 1/2$

$\rightarrow \dfrac{a(1-r^n)}{1-r} = \dfrac{2^{k-1}(1-(1/2)^n)}{1-1/2}$

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

$= 2^k - 1$

let $n-k = 0$

$\rightarrow \boxed{n = k}$

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

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

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

$T(n) = O(1)$

Q5. What shall be time complexity of

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

Signature..............................

STUDENT STYLE

$i = 1, 2 \quad 3 \quad 4 \quad 5 \quad 6 \ldots$

$s = 1 * 3 * 6 * 10 \ldots\ldots n$

Sum of $s = 1+3+6+10+\cdots + T_n$ — (1)

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

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

$T_k = 1+2+3+4+\cdots n - T_n$

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

for k iteration,

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

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

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

$O(k^2) <= n$

$O(k^2) <= n$

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

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

Q6 Time complexity of -

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

Signature...............................

Date :
Topic :

as $i^2 <= n$

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

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

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

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

$T(n) = O(n)$

Ans ↓

Q7. for $k = k*2$, $k = 1, 2, 4, 8 \dots n$

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

$\Rightarrow \frac{a(r^n - 1)}{r-1}$; $\frac{1(2^k - 1)}{1}$

$n \rightarrow 2^k$

$\log n = k$

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

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

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

Ans ↓

Q8. $O(1)$

$i = 1, 2, 3, 4 \dots n \rightarrow O(n)$

$j = 1, 2, 3, 4 \dots n^2 \rightarrow O(n^2)$

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

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

Signature..............................

STUDENT STYLE

Date :
Page :
Topic :

$c = \log_3 1 = 0$

$n^0 = 1 > (f(n) = n^2)$

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

Ans →

Q9.

for $i = 1 \to j = 1, 2, 3, 4 \dots n = n$

for $i = 2 \to j = 1, 3, 5 \dots n = n/2$

for $i = 3 \to j = 1, 4, 7 \dots n = n/3$

$\vdots$

for $i = n \to j = 1 \dots 1$

$\Rightarrow \sum_{j=n}^{1} n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \dots + 1$

$\sum_{j=n}^{1} n \left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \frac{1}{n}\right]$

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

$\to T(n) = [n[\log n]]$

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

Ans →

Q10. as given n^k and c^n

relation b/w n^k and c^n is

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

as $n^k \le ac^n$

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

for $n_0 = 1$

$c = 2$

$\to 1^k \le a 2^1$

$n_0 = 1$ and $c = 2$

Signature..................................

—— × ——

STUDENT STYLE