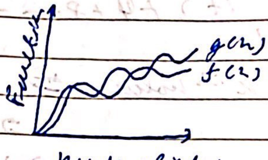
Name - Prabhdeep Singh
Section - G
Roll no - 04

Design and analysis of algorithm

Tutorial 1

Sol. Asymptotic notation - help you find the complexity of an algorithm when input is very large

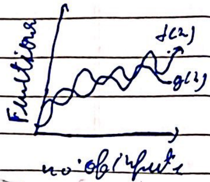
1) Big O



$f(n) = O(g(n))$
if $f(n) \le g(n)$
$\forall n \ge n_0$
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 $f(n)$



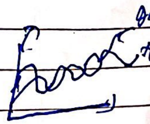
$f(n) = \Omega(g(n))$
if $f(n) \ge g(n)$
$\forall n \ge n_0$ for some constant $C > 0$.

3) Theta (θ)

$f(n) = \theta(g(n))$
$g(n)$ is both tight upper and lower bound for $f(n)$
$f(n) = \theta g(n)$
if
$c_1 g(n) \le f(n) \le c_2 \cdot g(n)$
$\forall n \ge (\max(n_1, n_2))$ for some constant $c_1 > 0$ and $c_2 > 0$

4) Small o (o)

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



Soln 2

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

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

GP K th value $\Rightarrow$ $T_K = a r^{K-1}$

$\Rightarrow 1 \times 2^{K-1}$

$\Rightarrow n = 2^{K-1}$

$2n = 2^K$

$\log 2n = K \cdot \log 2$

$\log 2 + \log n = K \log 2$

$\log n = K$

$O(K) = O(1 + \log n)$

$= O(\log n)$

Soln 3 $T(n) = \{3 T(n-1)$ 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

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

$T(n) = 9T(n-2)$ — 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 $n-K=0 \Rightarrow n=K$

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

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

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

classmate

Date

Page

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

$$\left.\begin{aligned} T(n) &= 2T(n-1) \\ T(n-1) &= 2T(n-2) \\ T(n-2) &= 2T(n-3) \\ T(1) &= 2T(0) \end{aligned}\right\} n \text{ levels}$$

$T(0) = 1$

Substituting all the values we get

$T(n) = 2^n \times T(0)$

$T(n) = 2^n \times 1$

$= O(2^n)$

Sol 5

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

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

$S = 1+3+6+10+15+\ldots$

Sum of $S = 1+3+6+10+\ldots+T_{n-1}+T_n$ — ①

also $S = 1+3+6+10+\ldots+T_n+\ldots$ — ②

from ① − ②,

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

$T_K = 1+2+3+4+\ldots K$

$T_K = \frac{1}{2}K(K+1)$

For K iteration, $1+2+\ldots+K \le n$

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

$O(K)^2 \le n$

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

Scanned with CamScanner

Classmate
Date
Page

Soln 6

```
void function (int n) {
  int i; count = 0;
  for (i=1; i*i <= n; i++)
      count++; //O(1)
  }
```

as $i^2 <= n$

$i <= \sqrt{n}$

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

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

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

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

$T(n) = O(n)$

Ans 7

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

for $k = K*2$

$K = 1, 2, 4, 8, \ldots n$

$\Rightarrow GP \Rightarrow a = 1, r = 2$

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

$= 1(2^K - 1)$

$n = 2^K \Rightarrow \log n = K$

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

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

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

Q.- function (int n)

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

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

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

$c = \log_3 1 = 0$

$\Rightarrow n^0 = 1 < (f(n) = n^2)$

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

9 void function (int n) {
for (i = 1 to n) { //O(n)
for (j = 1; j ≤ n; j = j + i)

printf ("*"); //O(1)
}

}

for $i = 1 \Rightarrow j = 1, 2, 3, 4 \ldots n = n$
for $i = 2 \Rightarrow j = 1, 3, 5, \ldots n = n/2$
for $i = 3 \Rightarrow j = 1, 4, 7, \ldots x = n/3$

for $i = n \Rightarrow j = 1 \ldots = 1$

$\Rightarrow \sum_{j=x} n + n/2 + n/3 + n/4 \ldots + 1$

$\Rightarrow \sum_{j=x} n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n}\right)$

$\sum_{j=n} n(\log n)$

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

Sol: 10 Relation b/w n^k and c^n is
$n^k = O(c^n)$
as $n^k \le a c^n$ ∀ $n \ge n_0$ and
some constant $a > 0$
for $n_0 = 1$
$c = 2$
$\Rightarrow 1^k \le a 2^1$
$n_0 = 1$ and $c = 2$

Scanned with CamScanner