PAGE _____
DATE _____

Ques) Asymptotic Notations : Mathematical Notations that are used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value.

OR

Those notations are used to tell the complexity of an algorithm when input is very large.

Big O Notations
We compute the big O of an algorithm by counting the number of iterations the algorithm always takes the input of n.

Big O Notation
The Big O notations describes the worst-case running time of an program. We compute the big-O of an algorithm by counting how many iterations an algorithm will take in the worst case scenerio with an input of N. For example, O(log n) describes the Big O of a binary search algorithm

Big Ω Notations
Big Ω (omega) notations describes the best case running time of program. We compute the big-Ω by counting how many iterations an algorithm will

PAGE
DATE

take in the best-case scenario based on an input of N. for example, a Bubble Sort Algorithm has a running time of $\Omega(N)$ because in the best case scenario the list already sorted, and the bubble sort will terminate after the first iteration.

4 Small O Notation
It is used to describe an upper bound that cannot be tight. In other words, loose upper bound of $f(n)$.

5 small omega Notation :
commonly written as ω, is an asymptotic notation to denote the lower bound (that is not asymptotically tight) on the growth rate of runtime of an algorithm

Ques2 for (i = 1 to N) { i = i * 2; }

$i = 1, 2, 4$

$i = \underbrace{2^0, 2^1, 2^2 \ldots 2^n}_{k \text{ terms}}$

This forms a GP, where

$a = 1$, $r = \frac{a_2}{a_1} = \frac{2}{1} = 2$

$t_k = ar^{k-1}$

$t_k = 1 \times 2^{k-1}$

PAGE _____
DATE _____

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

$$2n = 2^k$$

Taking log on both sides

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

$$k = 1 + \log_2 n$$

Time complexity $O(\log_2 n)$

$$\sum_{i=1 (i=i*2)}^{N} (1 + 1 + 1 \ldots \log_2 n)$$

$$= O(\log_2 n)$$

Ques3 $T(n) = 3T(n-1)$ — ①

put $n = n-1$ in ①

$T(n-1) = 3T(n-2)$ $\quad T(n) = 3(3T(n-2))$

$T(n) = 9T(n-2)$ — ②

put $n = n-2$ in ①

$T(n-2) = 3T(n-3)$ — ③

Substitute ③ in ②

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

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

PAGE
DATE

Put $n-k=0$

$k = n-0$

$$T(n) = 3^{n-0} T(0) = \frac{3^n}{3^0}$$

$$O(3^n)$$

Ques 4 → $T_n \begin{cases} 1 & n=0 \\ 2T(n-1)-1 & n>0 \end{cases}$

$$T(n) = 2T(n-1) - 1 \quad - (1)$$

put $n = n-1$ in (1)

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

$$T(n) = 4T(n-2) - 2 - 1 \quad - (2)$$

put $n = n-2$ in (1)

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

Substitute in (2)

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

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

$$T(n) = 2^k T(n-k) - \underbrace{(2^0 + 2^1 + 2^2 \cdot 2^{k-1})}_{k \text{ terms}}$$

$n-k=0 \quad n=k$

$$= 2^n T(0) - 1 \times \frac{(2^k - 1)}{2-1}$$

$$= 2^n - 2^k + 1$$

$$T(n) = 2^n - 2^n + 1 \qquad O(1)$$

PAGE
DATE

Ques 5
```
int i=1, s=1;
while (s<=n)
{  i++;          → O(1)
   s+=1;         → O(1)
   print("#");   → O(1)
}
```

S = 1, 3, 6, 10, 15 . . . .
(2, 3, 4, 5)

First difference is in AP

$T_n = Ak^2 + Bk + C$

putting $k = 1$

$A + B + C = 1 \quad -(1)$

putting $k = 2$

$4A + 2B + C = 3 \quad -(2)$

putting $k = 3$

$9A + 3B + C = 6 \quad -(3)$

solving (1) (2) and (3)

$A = \frac{1}{2} \quad B = \frac{1}{2} \quad C = 0$

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

$n < \frac{k(k+1)}{2}$ Time complexity $O(\sqrt{n})$

PAGE
DATE

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

for($\underbrace{i = 1}_{O(1)}$; $i * i <= n$; $i++$)

values of i

$$\underbrace{1, 4, 9, 16 \ldots (\sqrt{n})^2}_{K}$$

First difference of this series forms AP.

$$Ak^2 + Bk + C = t_k$$

Put $K = 1$

$$A + B + C = 1 \quad - (1)$$

Put $K = 2$

$$4A + 2B + C = 4 \quad - (2)$$

Put $K = 3$

$$9A + 3B + C = 9 \quad - (3)$$

Solving (1), (2) and (3)

$A = 1$, $B = 0$ and $C = 0$

PAGE
DATE

$n = AK^2 + BK + C$

$n = AK^2 + 0 + 0$

$n = K^2$

$K = \sqrt{n}$

Time complexity : $O(\sqrt{n})$

Ques7
```
void function(unit n)
{ unit 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 ++
}
```

i	j	k
$n/2$	$\log n$	$\log n \times \log n$
$\frac{n}{2}+1$	$\log n$	$\log n \times \log n$
.		
.		
.		
n	$\log n$	$\log n \times \log n$

$O(n * (\log_2 n)^2)$

PAGE
DATE

Ques 6

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

$\underbrace{(n-3), (n-6), (n-9), \cdots (1)}_{K}$

$a = n-3 \qquad d = -6 - (-3) = -3$

$1 = (n-3) + (k-1)(-3)$

$1 = (n-3) - 3k + 3$

$3k = n-1$

$k = \frac{n-1}{3} = O(n * n^2)$

$O(n^3)$

PAGE _____
DATE _____

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

For i=1, j= n times
i=2, j= n/2 times
i=3, j= n/3 times
.
.
.
i=K, j= n/K times

i=n, j= n/n times

Total Time Complexity: $n + n/2 + n/3 + \dots n/n$

$$n * \underbrace{(1 + 1/2 + 1/3 + \dots 1/n)}_{\log(n)}$$

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

$$= \sum_{K=1}^{n} \frac{1}{K}$$

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

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

PAGE
DATE

Question $f(n) = n^k \quad g(n) = c^n$

where $k >= 1$ and $c > 1$

let $k = 1$ & $c = 2$

$f(1) = (1)^1 \qquad g(1) = (2)^1$

$f(1) < g(1)$

$f(2) = (2)^1 \qquad g(2) = (2)^2 = 4$

$f(2) < g(2)$

satisfies O notation $f(n) \leq g(n)$

$f(n_0) = c_0 \cdot g(n_0)$

$n_0^k = c_0 \cdot c^{n_0}$

$k = 1, \quad c = 2$

$n_0^1 = c_0 \cdot 2^{n_0}$

$\left(\frac{n_0}{c_0}\right)^1 = (2)^{n_0}$

Comparing

$\boxed{n_0 = 1}$

$\frac{n_0}{c_0} = 2$

$\boxed{c_0 = \frac{1}{2}}$

PAGE _____
DATE _____

$f(n) \leq 0.5\, g(n)$

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