Aditya Bahl
20021038
B, 25

Date ___/___/___ Page No.: ___

Tutorial 1

Ans 1 - **Asymptotic Notations** : It is used to describe the running time of an algorithm - how much time an algorithm takes with a given input, 'n'. There are 3 different notations big O, big theta (Θ) & big omega (Ω)

i) **Big O** : It defines an upper bound of an algorithm. It counts how many iterations an algorithm will take in the worst-case scenario with an input N

ii) **Big Ω notation** : Big omega (Ω) notation describes best running time of algorithm. For eg :- Bubble sort algorithm

iii) **Θ Notation** : Theta notation enclose (f) from above & below. It represents upper & lower bound of algorithm. Used for calculating average-time complexity.

Ans 2 -

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

Loop runs upto n time
i.e 1 . . . . n
Inside loop : i = i * 2 i.e i = 1 * 2 (for first value)
i = 2
i = 4 (2nd value)
i = 8 (3rd value)

Date / / Page No:

$i = 1, 2, 4, 8, 16, \ldots n$

$= 1 \cdot 2^1, 2^2, 2^3, 2^4, \ldots 2^n$

$\rightarrow 2^k = n \rightarrow k = \log n$

$\therefore T(c) = O(n)$ i.e. $O(\log n)$

Ans 3 - $T(n) = \{3T(n-1) \text{ if } n>0\}$

$n = n-1$

$\Rightarrow T(n-1) = 3T(n-1-1)$

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

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

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

for $n = n-1$

$\rightarrow T(n) = 9T((n-1)-2)$

$T(n-1) = 9T(n-3)$

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

$= 27T(n-3)$

$\Rightarrow 3T(n-1), 9T(n-2), 27T(n-3) \ldots$

$\ldots 3^n T(n-n)$

$= 3^n T(0)$

$= 3^n$

$\Rightarrow T(L) = O(3^n) \cdot O(3^n)$

Ans 4 - $T(n) = [2T(n-1) - 1 \text{ if } n>0]$

$n = n-1$

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

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

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

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

Chitra

Date ___/___/_____ Page No.: _____

for $n = n-1$

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

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

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

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

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

$\cdots 2^n T(n-n) - 1$

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

$T(n) = 1$

$\Rightarrow T(c) = O(1)$

Ans 5 -

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

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

$\Rightarrow k^2 = n$

$\Rightarrow k = \sqrt{n} \rightarrow O(\sqrt{n}) = (c)$

Date ___/___/_____ Page No.: _____

Ans 6 -

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

$i = 1 \Rightarrow 1*1 <= n \Rightarrow 1 <= n$

for $i = 2 \; ; \; 1*2 < n \Rightarrow 2 <= n$

$1^{*} \; n \leq n \Rightarrow n \leq n$

$1, 2, 3 \ldots n \Rightarrow A.P$

Sum $\rightarrow \frac{n(n+1)}{2} = O(n^2)$

Ans 7 -

```
void function (int n)
{
int i, j, k, count = 0;
for (i = n/2 ; i <= n ; i++) ⟶ n/2
  for (j = 1 ; j <= n ; j = j*2) ⇒ log2 n
for (k = 1 ; k <= n ; k = k*2) ⇒ log2 n
  count ++;
}
```

$\Rightarrow T(c) = n \log_2^2 n$

Ans 8 -

```
function (int n)
{
}
if (n == 1)
  return ;
```

Date ___/___/_____ Page No.: _____

```
for (i = 1 to n) → O(n)
{

for (j = 1 to n) → O(n)
{
{
  print ("*");
}
}
  funtFun (n-3);
}
```

→ $T(n) = O(n^2)$

Ques 9 —

```
void funcFun (int n)
{

for (i = 1 to n)
{
  for (j = 1 ; j <= n ; j = j + i)
    print + ("+");
}
}
```

	1	2	3	. . .	n
i	1	2	3	. . .	(n-1)
j	2	3	4	. . .	|n|

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

Date ___/___/_____ Page No.: _____

Ans 10 - function - n^k , c^n

⇒ Relation b/w the two fl

∴ n^k = O (c^n) Ans