Page No
Date / 201

Assignment

Ans 1) Asymptotic means towards infinity (the size of the input is very large). These notations are used to tell the complexity of an algorithm.

Notations :-

1) Big O - Let $f(n) = O(g(n))$ $f(n) \le c \cdot g(n)$
means $g(n)$ is tight upper bound of $f(n)$, $f(n)$ can never go beyond $g(n)$.

2) Big Omega (Ω) :
$f(n) = \Omega(g(n))$ $f(n) \ge c \cdot g(n)$
$g(n)$ is tight lower bound of $f(n)$, $f(n)$ can never perform better than $g(n)$.

3) Theta (Θ) :
Theta gives both 'tight' lower and upper bound.

4) small - Oh (o) :-
o gives upper bound (not tight). $f(n) < c \cdot g(n)$

5) small - omega (ω) -
ω gives lower bound (not tight) $f(n) > c \cdot g(n)$

Ans 2)
```
for ( i=1 to n) {
    i = i*2
}
```
Time complexity $= O(\log n)$.

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

$T(2) = 3T(n-1) = 3$
$T(3) = 3T(2) = 9$
$T(4) = 27$

Time complexity $= 3 + 9 + 27 - - - - (n-1)^3$
$= 3^n.$
$= O(3^n).$

Ans 4) $T(n) = 2(T(n-1)-1)$
$T(n-1) = 2(T(n-2)-1)$
$T(n) = 4T(n-2) - 2 - 1$
$T(n-2) = 2T(n-3) - 1$
$T(n) = 8T(n-3) - 4 - 2 - 1$
$T(n) = 2^k \ldots\ldots 2^2 - 2^1 - 2^0$
Time complexity $= O(1)$

Ans 5)

S	i
1	1
3	2
6	3
10	4

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

Ans 6) $i * i = n$
$i = \sqrt{n}$
$TC = O(\sqrt{n})$

Ans 10) n^k is $O(c^n)$ as -
If $n = 2$ $k = 2$, $c = 2$

Then $2^2 < 2^2$

So, c^n is the upper limit of n^k.

Ans 11)

j = 1	i = 0
1	1
2	3
3	6
4	10

Time Comp = $O(2^n)$

Ans 12)

f(n) — 1

f(n-1)   f(n-2) — 2

f(n-2)   f(n-2)   f(n-3)   f(n-4) — $\cancel{2}^2$

⋮

2^n

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

Ans 13) nlogn

```
for (i=0 to n)
  for (j=0 to n ; j=j*2)
    c++;
```

* n^3

```
for (i=0 to n)
  for (j=0 to n)
    for (k=0 to n)
      c++,
```

- Log (logn)

```
int func (int n)
    if (n==1)
        return n;
    else
        return fun(√n) + fun(√n)
    }
```

Ans 14) $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + Cn^2$

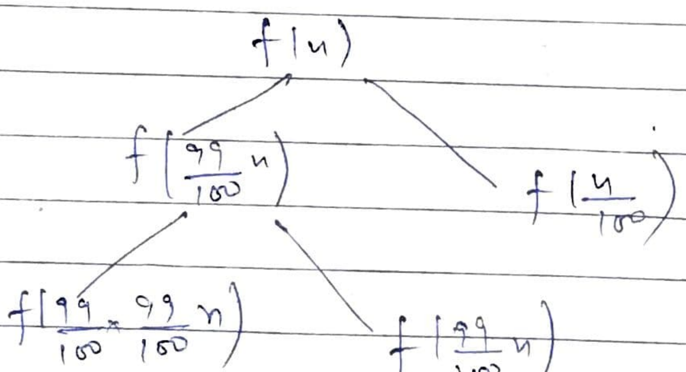
Using Master theorem.

$a = 2 \quad b = 2 \quad c = 1$

$f(n) \geq n^2 \qquad f(n^2) \ngtr 1$

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

Ans 17) $T(n) = T\left(\frac{99}{100}n\right) + T\left(\frac{n}{100}\right)$



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

Ans 18.) a) $100 < \log(\log(n)) < \log(n) < \sqrt{n} < n < \log(n!)$
$< n\log n < n^2 < 2^n < 2^{2n} < 4^n < n!$

b) $1 < \log(\log n) < \sqrt{\log n} < \log 2^n < \log n < 2\log(n) <$
$n\log(n) < n < 2n < 4n < n^2 < \log(n!) < 2\cdot(2^n) < n!$

c) $96 < \log_2 n < \log_8 n < \log 5n < \log n! < n \log_6 (n) < n \log_2 (n)$
$< 18n^2 < 7n^3 < 8^{2n} < n!$

Ans 19) linear (arr, key)

```
{
    for (int i=0 ; i<n ; i++)
        if (arr[i] == key)
            return i;

    return -1;
}
```

Ans 21)

Algo	Best	Avg	Worst	Space.Com
Bubble	$O(n^2)$	$O(n^2)$	$O(n^2)$	1
Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$	1
Insertion	$O(n)$	$O(n)$	$O(n^2)$	1

Ans 24) $T(n) = T\left(\frac{n}{2}\right) + 1$