DATE □□□□□□□□

PAGE □□□

Design and Analysis of Algorithms

Assignment - 1

Ques-1 What do you understand by Asymptotic notations. Define different asymptotic notation with examples.

Ans 1-> Asymptotic notations are languages that allow us to analyze an algorithm running time by identifying its behaviour as the input size of the algorithm.

Types:-

1. Big O:- It is commonly used for worst case, and gives us upper bound for the growth rate of runtime of algorithm.
   Example:- Big O notation for linear search is $O(n)$

2. Big Omega:- It is notation used for least case complexity, it provides us with an symptotic lower bound.
   Ex - Big omega of linear search is $\Omega(1)$

3. Theta:- It is used for tight bound on the growth rate of runtime of algo
   Ex - Theta of linear search is $O(n)$

4. Small Omega:- It is used to denote the upper bound (i.e. not ~~g~~ asymptotic tight)
   $f(n) = o(g(n)) \triangleright f(n) < c(g(n)) \quad c > 0$

Teacher's Signature..........................

5. Small Omega - To denote lower bound (that is not asymptotic)

Ques-2 Time complexity of - for (i=1 to n) { i = i*2 }

=) O(log n)

Ques-3 T(n) = { 3T(n-1) if n>0 otherwise 1 }

$T_n = 3T(n-1)$

$T(1) =$ ~~$3T(1-1) = 3$~~ 1

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

$T(3) = 3T(9) = 27$

$T(4) = 3T(3) = 27$

!

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

Time complexity → $O(3^n)$

Ques-4 T(n) = { 2T(n-1) - 1 if n>0 otherwise 1 }.

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

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

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

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

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

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

$T(n) = 16T(n-4) - 8 - 4 - 2 - 1$

$T(n) = 2^k - \dots - 2^3 - 2^2 - 2^1 - 2^0$

TC = O(1)

Ques-5

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

i	S	n
~~1~~	~~1~~	10
~~2~~	3	
3	~~6~~	
4	10	

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

Ques-6

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

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

Ques-7

```
void function (int n){
    int i, 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++;
 }
```

Ans- $O(n\log_{2}^{2} n)$

DATE □□□□□□□□

PAGE □□□

Ques-9

```
void function (int n){
    for(i=0 to n)
        for(j=1;j<=n;j=j+1){
            printf("*")
}
```

Ans→ Total time complexity of problem is.

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

Ques-10 for the functions, n^k and a^n, what is the asymptotic relation between these functions?

Assume that $K>=1$, and $a>1$ are constants.

find out the value of c and no for which relation holds.

Ans n^k is $O(c^n)$ as for example

If we take $n=2$, $K=2$, $c=2$

Then $2^2 \leq 2^2$ so c^n is upper limit of n^K