PAGE No.
DATE: / 202

DESIGN ANALYSIS OF ALGORITHM

TUTORIAL - 2

1. What is the time complexity of below code?

```
void func (int n)
{
int j=1, i=0;
while (i<n)
 {
   i = i+j;
    j++;
 }
}
```

series = 1 + 2 + 3 + 4 + . . . < n

$\frac{k(k+1)}{2} < n$

$\frac{k^2+k}{2} < n$

$\therefore \quad k < \sqrt{n}$

$\therefore O(\sqrt{n})$

2. Write recurrence relation for the recursive function that prints Fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity and why?

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

T(n) — 1

T(n-1) T(n-2) — 2

T(n-2) T(n-3) T(n-3) T(n-4) — 4

T(n-3) T(n-4) T(n-4) T(n-5) T(n-4) T(n-5) T(n-5) T(n-6)

$$T(n) = 1 + 2 + 4 + 8 \dots$$
$$= 2^0 + 2^1 + 2^2 + 2^3 \dots 2^k$$

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

$$= O(2^n)$$ Time complexity

For space complexity the height of the tree determines the space assigned
∴ space complexity = O(n)

Q3 a) O(n (log n))

```
int i, j, sum = 0;
  for (i = 0; i < n; i = i*2)
     for (j = 0; j < n; j++)
        sum = sum + 1;
```

b) O(n^3)

```
int x, y, sum = 0, z;
  for (x = 0; x < n; x++)
     for (y = 0; y < n; y++)
        for (z = 0; z < n; z++)
           sum = sum + 1;
```

c) $\log(\log n)$

```
int func(int n)
{ int sum = 0, i;
  for (i = 0; i<n ; i = i*i)
        sum = sum+1;
}
```

4 Solve the following recurrence relation

$T(n) = T(n/4) + T(n/2) + cn^2$

we will ignore the lower order term

$\therefore T(n) = T(n/4) + cn^2$

$c = \log_4 1 \qquad c = 0$

$cn^2 > n^0$

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

5 What is the time complexity of following func.?

```
int func (int n)
{
  for (int i = 1; i <= n ; i++)
    {
      for (int j = 1 ; j<n; j+=1)
        {
          // some O(1) task
        }
    }
}
```

when $i = 1$

j loop executes for values

$1, 2, 3 \dots n-1$ — $(n-1)$ times

when $i = 2$

j loop executes for values

$1, 3, 5 \dots (n-1)$ — $\frac{n}{2}$ times

similarly $\frac{n}{3}$ times for $i = 3$

and $\frac{n}{4}$ times for $i = 4$

$$\therefore T(n) = n \underbrace{\left(1 + \frac{1}{2} + \frac{1}{3} \dots \frac{1}{n}\right)}_{}$$

$n \quad \log n$

$\therefore O(n \log n)$

PAGE No
DATE : / 202

6) What should be the complexity of the following function?

```
for (int i = 2; i <= n; i = pow(i, k))
{
    // some O(1) expressions
}
```

where k is constant

The loop executes for values

$2, 2^{k}, 2^{k^2}, 2^{k^3}, 2^{k^4} \ldots$

let it execute t times

$\therefore$ t th value $2^{k^t} = n$

$$\log_2 2^{k^t} = \log_2 n$$

$$k^t = \log_2 n$$

$$\log_k k^t = \log_k (\log_2 n)$$

$$t = \log_k (\log_2 n)$$

$$\therefore O(\log(\log n))$$

7.

7. Write a recurrence relation when quick sort repeatedly divides the array into two parts of 99% and 1%. Derive time complexity in this case. Show the recursion tree while deriving time complexity and find the difference in heights of both the extreme parts. What do you understand by this analysis?

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

$T(n)$ — (n)

$T\left(\frac{99n}{100}\right)$ $\quad$ $T\left(\frac{n}{100}\right)$ — (n)

$T\left(\left(\frac{99}{100}\right)^2 n\right)$ $\quad$ $T\left(\frac{99}{(100)^2} n\right)$ $\quad$ $T\left(\frac{99}{100^2} n\right)$ $\quad$ $T\left(\frac{n}{100^2}\right)$

└ (n)

more height

max height $= \frac{n}{(100/99)^k} = 1$

$n = \left(\frac{100}{99}\right)^k$

$\log_{100/99} n = k$

less height

$\hookrightarrow \leq (n)$

min height

$= \frac{n}{(100)^k} = 1$

$\log_{100} n = k$

for time complexity we take max. height and multiply it by the cost ~~of each~~ tree i.e n $\therefore$ $n \log_{100/99} n$

$= o(n \log n)$

PAGE No
DATE : / 202

difference in heights of extreme parts

$$\log_{100/99} n - \log_{100} n$$

$$\frac{\log n}{\log(100/99)} - \frac{\log n}{\log 100}$$

PAGE No
DATE: / 202

Q8 Arrange the following in increasing order of rate of growth.

a) n, $n!$, $\log n$, $\log\log n$, $\sqrt{n}$, $\log(n!)$, $\log^2 n$, 2^n, 2^{2^n}, 4^n, n^2, 100

$100 < \log\log n < \log n < (\log n)^2 < \sqrt{n} < n < n\log n < \log(n!) < n^2 < 2^n < 4^n < n! < 2^{2^n}$

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

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

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

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