SHRADDHA BAHUGUNA
CST-SPL1
30
2017032

Date
Page No.

DESIGN AND ANALYSIS OF ALGORITHMS

TUTORIAL-2

Q1 What is the Time Complexity of below code?

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

j	i
1	0
1	1
2	3
3	6
4	10
5	15

$S = 0+1+3+6+10+15+\ldots T_k$ —(1)

also $S = 0+1+3+6+\ldots T_{k-1}+T_k$ —(2)

$0 = 1+2+3+4+\ldots k - T_k$

$T_k = 1+2+3+4+\ldots k$

$T_k = \frac{1}{2} k(k+1)$

shraddha

Date.
Page No.

for k iteration

$$1+2+3+\ldots k < n$$

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

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

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

$$k \cong O(\sqrt{n})$$

$$\boxed{T(n) = O(\sqrt{n})}$$

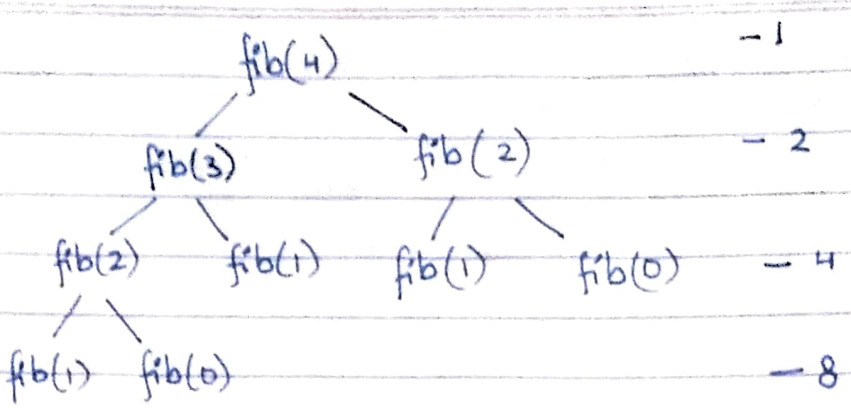
Q2 Write recurrence relation for recursive function that prints fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity of this program and why?

0 1 1 2 3 5 .. n

```
int fib(int n)
{
    if (n <= 1)
        return n;                          //O(1)
    return fib(n-1) + fib(n-2) - T(n-1) + T(n-2)
}
```

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

Shraddha



$$T(n) = 1+2+4+8+\ldots+n$$

$$S(n) = \frac{a(r^{n}-1)}{r-1}$$

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

$$T(n) = 2^{n}.2 - 1$$

$$\boxed{T(n) = O(2^{n})}$$

Space Complexity $\longrightarrow$ $O(1)$

As recursive implementation doesn't store any values and calculates every value from scratch
So, as complexity of 1 call is $O(1)$

$\therefore$ Total Space Complexity $= O(1)$

Shraddha

Date
Page No.

Q3 Program which have complexity.

1) $(n \log n)$

```
for (i=1 ; i<=n ; i=i*2)                    // log n
    for (j=1 ; j<=n ; j++)                  // n
        int s=1;
```

$O(n \log n)$

2) n^3

```
for (i=0 ; i<=n ; ++i)                      //n
    for (j=0 ; j<=n ; ++j)                  //n
        for (k=0 ; k<=n ; ++k)              //n
            cout << "Hi";
```

$O(n^3)$

3) $\log(\log n)$

```
for(int i=2 ; i<n ; i=pow(i,2)
        cout << "Hi";
```

Shraddha

Date.
Page No.

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

by neglecting lower order term $T(n/4)$

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

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

$$c = \log_2 1$$
$$= 0$$
$$n^c = n^0 = 1 \quad < cn^2$$

$$\boxed{T(n) = \theta(n^2)}$$

5
```
int function(int n)
{
    for (int i=1 ; i<=n ; ++i)
        for (int j=1 ; j<n ; j+=i)
            cout << "Hi" ;
```

for $i = 1$ $\quad j = 1 + 2 + 3 + 4 + 5 + 6 + \ldots$

for $i = 2$, $\quad j = 1, 3, 5, 7, \ldots n$.

for $i = 3$, $\quad j = 1, 4, 7, \ldots n$

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \ldots 1$$

$$n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots \frac{1}{n}\right)$$

$$= n \int_1^n 1/x$$

$$\boxed{O(n \log n)}$$

Shraddha

Date.
Page No.

6 Time Complexity

```
for(int i=2; i<=n; i=pow(i,k))
    {   }
```

where k is constant

first iteration $i = 2$

second $i = 2^k$

third $i = (2^k)^k = 2^{k^2}$

⋮

nth iteration $i = (2^k)^i = n$

$n = 2^{k^i} = n$

applying log $\rightarrow \log n = \log_2 k^i = k^i$

applying log again

$i = \log_k \log(n)$

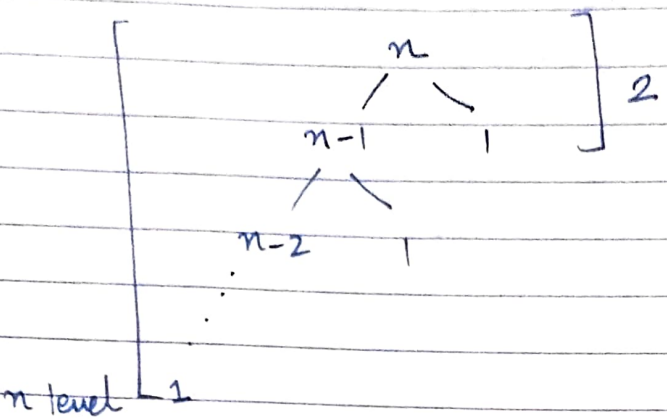
$$\boxed{T(n) = \log_k \log_2 n}$$

Shraddha

Date
Page No.

7 Given algo divides array in 99% and 1% part

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



$$n = [T(n-1) + T(n-2) + \dots T(1) + O(1)] \times n$$

$$= n \times n$$

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

lowest height = 2
highest height = n

$$\because \boxed{\text{diff} = n-2} \quad (\because n > 1)$$

The given algo provides linear result

Shraddha

Date
Page No.

8 Arrange the following in increasing order of growth rate.

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

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

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

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

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

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

Shraddha