Tutorial - 2

Name - Dhruv Goyal
Section - F
Roll No - 54
University Roll No - 2016729

Q1.) What is time complexity of below code & how?

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

j=1	i=1	n - level
j=2	i=1+2	
j=3	i=1+2+3	

for (i)

$\because 1+2+3+\ldots+m<n$

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

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

$m \approx \sqrt{n}$

By summation method

$\sum_{i=1}^{m} 1 = 1+1+\ldots\ldots+\sqrt{n}$ times

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

Q2) Write recurrence relation for function that prints Fibonacci series. Solve it to get the time complexity. What will be space complexity & why?

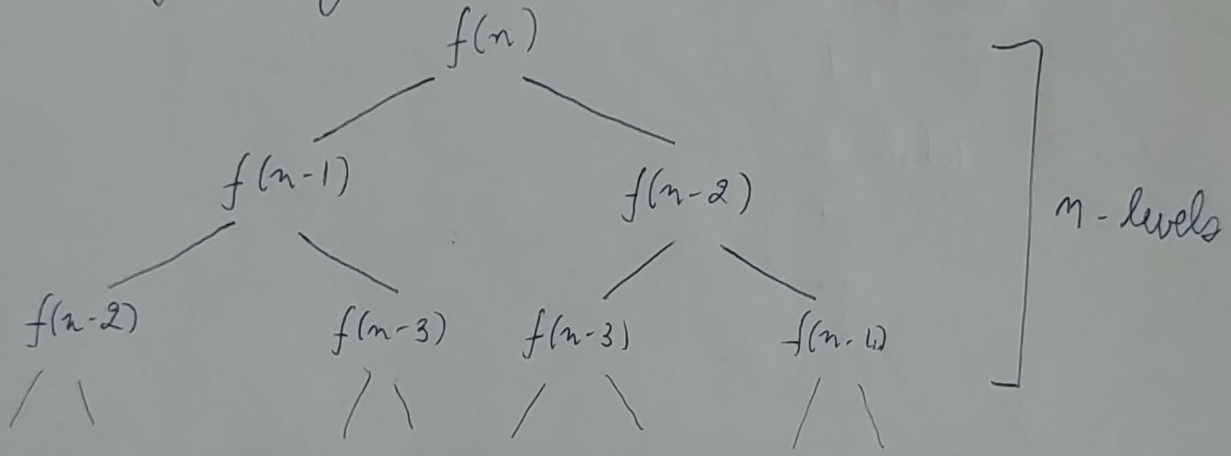
For Fibonacci Series -

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

$f(0) = 0$

$f(1) = 1$

By forming tree



∵ At every function call we get 2 function calls

∴ for n levels

We have = $2 \times 2 \ldots$ n times

$T(n) = 2^n$

Maximum Space -

Considering Recursive Stack:

No of calls maximum = n

For each call we have space complexity O(1)

∴ $\boxed{T(n) = O(n)}$

Without considering Recursive stack:

Each call we have time complexity O(1)

∴ $\boxed{T(n) = O(1)}$

13.) Write a program which have complexity
$n(\log n)$, n^3, $\log(\log n)$

1.) $n \log n$

```
void main(){
    int i, j; int n;
  for(i=0; i<n; i++)
    for(j=1; j<n; j*=2)
      printf("*");
}
```

2.) n^3

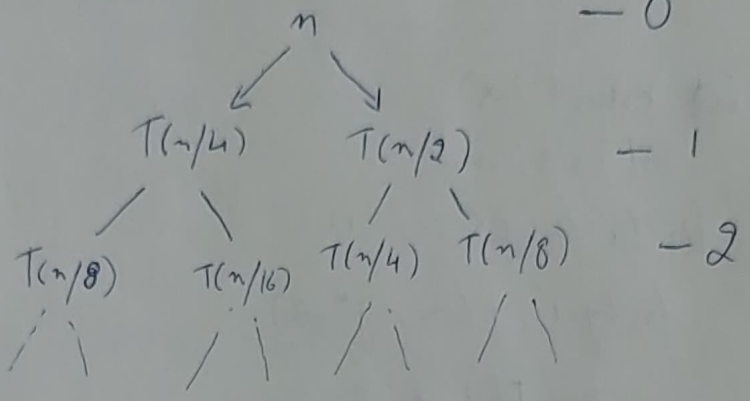
```
void main(){
  int i, j, k; int n;
  for(i=0; i<n; i++)
  for(j=0; j<n; j++)
  for(k=0; k<n; k++)
    printf("+");
}
```

3.) $\log(\log n)$

```
void main()
{ int count=0; int i;
  for(i=2; i<n; i=i*i)
  { count++;
  }
}
```

Q4.) Solve the following recurrence relation

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



— 0

— 1

— 2

At level →

$0 \rightarrow cn^2$

$1 \rightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{c5n^2}{16}$

$2 \rightarrow \frac{n^2}{8^2} + \frac{n^2}{16^2} + \frac{n^2}{4^2} + \frac{n^2}{8^2} = \left(\frac{5}{16}\right)^2 n^2 c$

.

.

.

$$\text{last level} = \frac{n}{2^k} = 1$$

$$K = \log_2 n$$

$$T(n) = c\left(n^2 + \left(\frac{5}{16}\right)n^2 + \left(\frac{5}{16}\right)^2 n^2 + \cdots + \left(\frac{5}{16}\right)^{\log n} n^2\right)$$

$$T(n) = cn^2\left[1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \cdots + \left(\frac{5}{16}\right)^{\log n}\right)$$

$$= cn^2 \times 1 \times \left(\frac{1 - (5/16)^{\log n}}{1 - 5/16}\right)$$

$$= cn^2 \times \frac{11}{5} \times \left(1 - \left(\frac{5}{16}\right)^{\log n}\right)$$

$$T(n) = O(n^2 c)$$

$$O(n^2 c)$$

5) What is time complexity of following func()?

```
int fun(int n){
    for(int i=1; i<=n; i++){
    for(int j=1; j<=n; j+=i){
    // Some O(1) tasks
    }}
}
```

for	i	j
	1	1
	2	1+3+5
	3	1+4+7
	⋮	⋮
	n	1+5+9

$j = (n-1)/i$ times

$$\sum_{i=1}^{n} \frac{(n-1)}{i}$$

$$\therefore T(n) = \frac{(n-1)}{1} + \frac{(n-1)}{2} + \frac{(n-1)}{3} + \cdots \frac{(n-1)}{n}$$

$$T(n) = n\left[1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right] - 1\left[1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right]$$

$$= n\log n - \log n$$

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

Q6) What should be time complexity of

```
for(int i=2; i<=n; i = pow(i,K))
    { // Some O(1)
    }
```

Where K is constant.

→ for

i
2^1
2^k
2^{k^2}
2^{k^3}
⋮
2^{k^m}

where

$2^{k^m} <= n$

$K^m = \log_2 n$

$m = \log k \log_2 n$

$\sum_{i=1}^{m} 1$

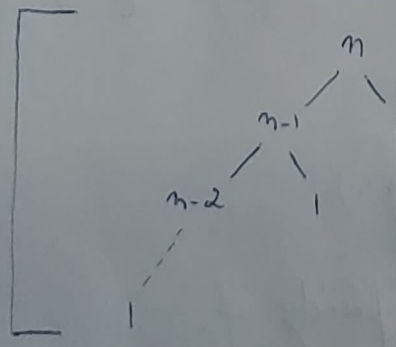
$1+1+1+ \ldots + m$ times

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

Q7) Write a recurrence relation when quick sort repeatedly divides array into 2 parts of 99% & 1%. Derive time complexity in this case. Show the recurrence tree while deriving time complexity & find difference in heights of both extreme parts. What do you understand by this analysis?

Given algorithm divide away in 99% & 1% part.

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



n - levels

'n' work is done at each level

$T(n) = (T(n-1) + T(n-2) + \ldots + T(1) + O(1)) \times n$

$= n \times n$

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

lowest height = 2

highest height = n

$\therefore$ **difference = n − 2** $\quad n > 1$

The given algorithm produces linear result

Q8) Arrange following in increasing order of rate of growth:

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

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

b) $2(2^n), 4n, 2n, 1, \log n, \log(\log n), \sqrt{\log n}, \log 2n, 2\log n, n, \log(n!), n!, n2, n\log n$

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

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

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