Tutorial - 2

Q1. what is time complexity of following code:

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

→ 1st time $i = 1$

2nd time $i = 1 + 2$

3rd $i = 1 + 2 + 3$

4th = $i = 1 + 2 + 3 + 4$

⋮

For k time $i = (1 + 2 + 3 + \cdots k) < n$

$= k(k+1)/2 < n$

$= k^2 < n$

$k = \text{sqrt}(n)$

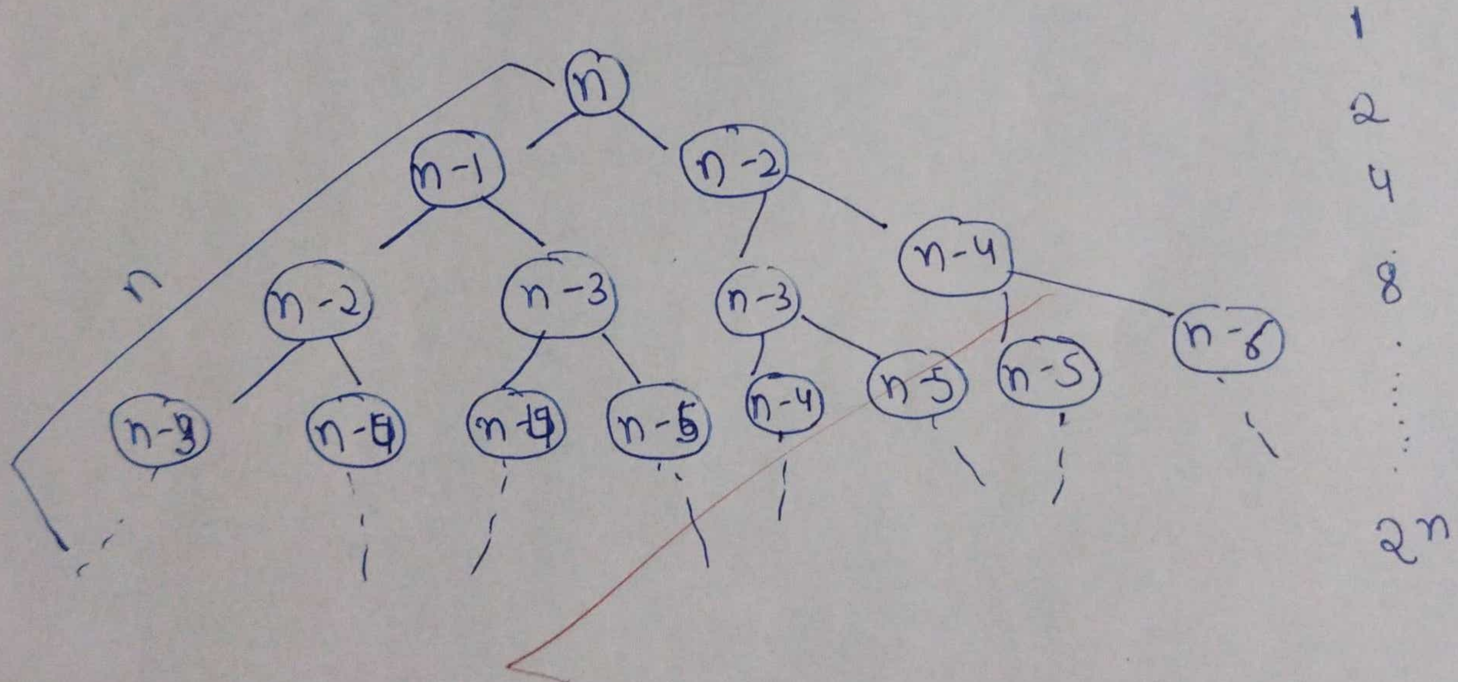
∴ time complexity = $O(\text{sqrt}(n))$

Q2. Write recurrence relation for the recursive function that print Fibonacci series.

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

Solving by tree method:

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



Complexity : $1+2+4+\dots 2^n = 2^0+2^1+2^2+\dots 2^n$

$a=1, r=2$

Using GP ⇒

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

Q13. Write programs which have complexity

- n(log n) :-

```
main ()
{ int i, j;
for (i=1; i<=n; i++)  //n
for (j=1; j<=n; j*=2)  //log n
  sum+=j; }
```

- n^3 :-

```
main ()
{ int i, j;
for( i=1; i<=n; i++ )  //n
for (j=1; j<=n; j+=2)  //n
for (k=1; k<=n; k++)  //n
  sum+=k; }
```

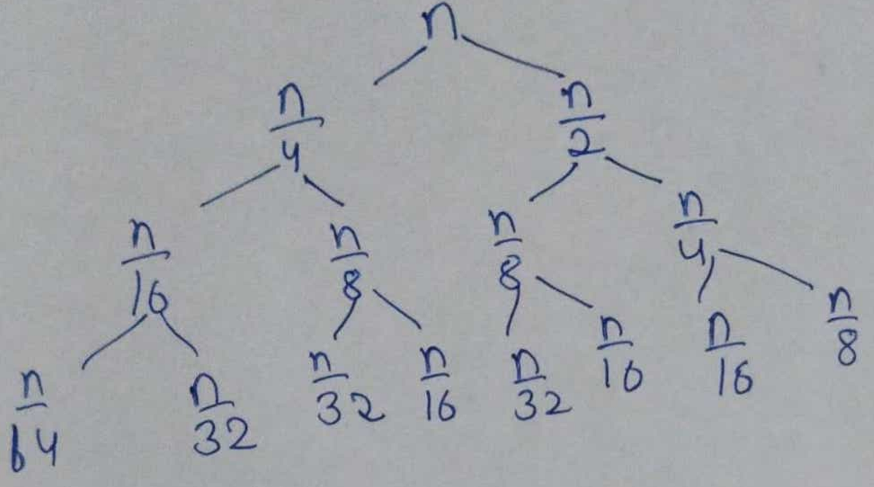
- log (log n)

```
for (i=2; i<n; i=i*i)
  { count++; }
```

Q4. Solve following reccurrence relation

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

$\rightarrow$



2^0

2^1

2^2

2^3

$\vdots$

2^k

$n = 2^k$

$k = \log_2 n$

$0 \rightarrow cn^2$

$1 \rightarrow \frac{5}{16} cn^2$

$2 \rightarrow \left(\frac{5}{16}\right)^2 cn^2$

$3 \rightarrow \left(\frac{5}{16}\right)^3 cn^2$

$\vdots$

$k \rightarrow \left(\frac{5}{16}\right)^k cn^2$

$\therefore cn^2 \left[\left(\frac{5}{16}\right)^0 + \left(\frac{5}{16}\right)^1 + \left(\frac{5}{16}\right)^2 + \dots \dots \left(\frac{5}{16}\right)^R \right]$

$a = 1, \; r = \frac{5}{16} < 0$

$$T(n) = a\frac{(1-r^R)}{1-r} \cdot cn^2 = 1\frac{\left(1-\left(\frac{5}{16}\right)^R\right)}{1-\frac{5}{16}} cn^2 = \frac{1-\left(\frac{5}{16}\right)^R}{\frac{11}{16}} cn^2$$

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

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

$$= \frac{16}{11} cn^2$$

$\therefore$ Time complexity $= \Theta(n^2)$

Q5. What should be time complexity of

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

$\rightarrow$ $i = 1 \Rightarrow j = 1, 2, 3, 4, \dots\dots n = n$

$i = 2 \Rightarrow j = 1, 3, 5, 7, \dots\dots n = n/2$

$i = 3 \Rightarrow j = 1, 4, 7, 10, \dots\dots n = n/3$

$\vdots$

$i = n \Rightarrow j = n/n$

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \dots\dots 1$$

$$a = n\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots 1\right)$$

$\Rightarrow a = 1, \ r = \frac{1}{2}$

$$= \frac{a(1-r^n)}{1-r} \cdot n = \frac{n\left(1-\left(\frac{1}{2}\right)^n\right)}{1-\frac{1}{2}} = n(2^n - 1)$$

$$= n(\log n - 1)$$

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

Q6: What should be time complexity of

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

// some O(1)

where k = constant.

$\rightarrow$ for $i = 2^1, 2^k, 2^{k^2}, 2^{k^3}, \dots\dots 2^{k^m}$

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

$k^m <= \log_2 n$

$m = \log_k \log_2 n$

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

$\Rightarrow 1 + 1 + 1 + \dots$ m times

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